

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ "РОЗУМНИЙ БУДИНОК"
З ВИКОРИСТАННЯМ ДАТЧИКІВ РУХУ ТА ТЕМПЕРАТУРИ DESIGN OF AN
INTELLIGENT SMART HOME SYSTEM WITH INTEGRATED MOTION AND
TEMPERATURE MONITORING

Виконав(ла): студент(ка) 4 курсу, групи КНТ-213сп
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

МІЩЕНКО І.С.

(ПРИЗВИЩЕ та ініціали)

Керівник СТЕПАНЕНКО О.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент ДЬЯЧУК Т.С.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
 Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2026 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

МІЩЕНКА Івана Сергійовича
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення інтелектуальної системи "Розумний будинок"
з використанням датчиків руху та температури
Design of an Intelligent Smart Home System with Integrated Motion and Temperature
Monitoring

керівник проєкту (роботи) к.т.н., доцент, СТЕПАНЕНКО Олександр Олексійович

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 07 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 10 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.

2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	СТЕПАНЕНКО О.О., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання «20» квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Іван МІЩЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Олександр СТЕПАНЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
109 с., 4 табл., 46 рис., 3 дод., 23 джерел.

РОЗУМНИЙ БУДИНОК, ДОМАШНЯ АВТОМАТИЗАЦІЯ, ЧАТ-БОТ, TELEGRAM BOT API, HOME ASSISTANT, RASPBERRY PI, ДАТЧИКИ МІКРОКЛІМАТУ, ІНТЕРНЕТ РЕЧЕЙ (IoT).

Об'єкт дослідження – методи і алгоритми та їх застосування для створення програмної системи віддаленого моніторингу на основі Інтернету речей.

Предмет дослідження – методи, архітектурні рішення та програмно-апаратні засоби інтеграції локального сервера домашньої автоматизації Home Assistant з месенджером Telegram для забезпечення дистанційного контролю.

Мета роботи – підвищення комфорту, безпеки та ефективності керування параметрами мікроклімату приміщення шляхом розробки і налаштування системи «Розумний будинок» з інтегрованим кросплатформним чат-ботом Telegram.

Матеріали, методи та технічні засоби: одноплатний комп'ютер Raspberry Pi 4 (як центральний сервер керування) , операційна система Home Assistant OS, бездротові датчики температури, вологості та відчинення дверей екосистеми Tuuya Smart , мережевий протокол Wi-Fi (IEEE 802.11 b/g/n), мова розмітки конфігурацій YAML , інтерфейси взаємодії Telegram Bot API та Tuuya IoT Platform.

Результати. Розгорнуто та сконфігуровано локальний сервер автоматизації на базі Raspberry Pi 4. За допомогою хмари Tuuya IoT інтегровано бездротові датчики телеметрії. Створено та підключено інтерактивного Telegram-бота з меню інлайн-кнопок, що забезпечує отримання поточних

показників мікроклімату та миттєвих тривожних сповіщень про стан охорони (відчинення дверей) у режимі реального часу.

Висновки. Розроблена програмно-апаратна система дозволяє автоматизувати побутові процеси, забезпечує цілодобовий автономний контроль за станом приміщення та гнучкість розширення периферійних пристроїв без зміни базового ядра архітектури.

Галузь використання – системи автоматизації житлових та офісних приміщень, цифрові рішення для Smart Home та IoT-інфраструктур.

ABSTRACT

Explanatory note to the bachelor's diploma qualification work: 109 pages, 4 tables, 46 figures, 3 appendices, 23 sources.

SMART HOME, HOME AUTOMATION, CHATBOT, TELEGRAM BOT API, HOME ASSISTANT, RASPBERRY PI, CLIMATE SENSORS, INTERNET OF THINGS (IoT).

The object of research is methods and algorithms and their application for creating a remote monitoring software system based on the Internet of Things.

Subject of study are methods, architectural solutions, and software-hardware tools for integrating the local home automation server Home Assistant with the Telegram messenger to provide remote control.

The purpose of the work is to enhance the comfort, security, and management efficiency of indoor microclimate parameters by developing and configuring a "Smart Home" system with an integrated cross-platform Telegram chatbot.

Materials, methods, and tools: Raspberry Pi 4 single-board computer (as the central management server) , Home Assistant OS , wireless temperature, humidity, and door opening sensors from the Tuya Smart ecosystem , Wi-Fi network protocol (IEEE 802.11 b/g/n) , YAML configuration markup language , Telegram Bot API, and Tuya IoT Platform interfaces.

Results. A local automation server based on Raspberry Pi 4 has been deployed and configured. Wireless telemetry sensors have been integrated using the Tuya IoT cloud platform. An interactive Telegram bot featuring an inline button menu has been created and connected, providing real-time access to current microclimate metrics and instant security alerts (door status).

Conclusions. The developed software and hardware system automates routine household processes, ensures 24/7 autonomous monitoring of premises, and

offers scalable expansion of peripheral devices without requiring modifications to the core architecture.

Field of use is the automation of residential and commercial spaces, and digital solutions for Smart Home and IoT infrastructures.

ЗМІСТ

ВСТУП.....	11
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1. Актуальність «Розумного будинку».....	12
1.2. Етапи розробки системи «Розумний будинок».....	12
1.3 Аналіз існуючих програмних рішень та технологічних аналогів.....	14
1.4 Визначення цілей та завдань системи.....	16
1.5 Висновок за розділ 1	17
2 ВИБІР ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ	18
2.1 Популярні програмні засоби	18
2.2 Home Assistant:	20
2.3 Telegram.....	22
2.4 Tuuya Smart	26
2.5 Вибір апаратного забезпечення для реалізації інформаційної системи	28
2.6 Raspberry Pi	30
2.7 Датчики	30
2.8 Wi-fi	31
2.9 Висновок за розділ 2	32
3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ПРОГРАМНИХ КОМПОНЕНТІВ СИСТЕМИ	33
3.1 Об'єктно-модульна структура програмного забезпечення системи.....	33
3.2 Обґрунтування стеку системних бібліотек та фреймворків	36
3.3 Розробка прикладного протоколу та структури пакетів даних.....	38
3.4 Організація інформаційного обміну за допомогою технології Webhooks	39
4 НАЛАШТУВАННЯ І ТЕСТУВАННЯ СИСТЕМИ	42

4.1 Розробка та налаштування системи	42
4.2 Налаштування системи.....	43
4.4 Підключення датчиків до НА	58
4.5 Створення чат-боту	61
5 КЕРІВНИЦТВО КОРИСТУВАЧА	64
5.1 Налаштування конфігурації НА	64
5.2 Автоматизація служб	69
5.3 Безпека системи НА	78
ВИСНОВКИ.....	80
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	81
ДОДАТОК А Технічне завдання	83
ДОДАТОК Б Фрагмент коду програми	90
ДОДАТОК В Слайди презентації.....	99

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

BF	-	BotFather
IP	-	Internet Protocol;
HA	-	Home Assistant
ДП	-	Дипломний проєкт;
ПЗ	-	Пояснювальна записка;
ПК	-	Персональний комп'ютер;
ТЗ	-	Технічне завдання

ВСТУП

Щодня з'являється все більше і більше гаджетів, здатних зробити керування будинком простіше та комфортніше, тому будинки з великою кількістю розумної техніки стали називати «розумними». Розумний дім — це мережа датчиків і техніки, об'єднаних в єдину систему і підтримка керування та налаштування зі смартфона, планшета, комп'ютера або вбудованої сенсорної панелі. Обов'язковим елементом «Розумного будинку» є центр керування, він підтримує зв'язок з іншими пристроями, отримує від них інформацію, яку потім передає власнику за допомогою різних додатків. Техніка підключається через Wi-Fi та передає сигнали на ваш смартфон[1].

Система «розумний дім» допомагає автоматизувати певні побутові процеси, урізноманітнити дозвілля. Навіть найпростіший за проектом дім можна доповнити певним прогресивним обладнанням, яке розширить функціональні можливості житлової площі та усучаснить пересування. За допомогою технологій інтелектуального будинку, піч, наприклад, може повідомити господарів, коли вона потребує чистки, або коли холодильнику стане необхідний техогляд, він «скаже» про це. Сигналізація може одночасно подзвонити на номери служби безпеки та господаря будинку, якщо у домі з'явився незваний гість. За допомогою налаштувань мульти-рум, будинок може визначити, хто із членів родини пересувається по помешканню, і включити таке освітлення (температуру/музику тощо), яке влаштовує саме цю людину. Користувач може надати право доступу друзям та членам сім'ї за допомогою спеціально згенерованого ключа. Кожного разу, коли хто-небудь відкриває двері, власник буде отримувати повідомлення на телефон[2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1. 1 Актуальність «Розумного будинку»

На сьогоднішній день система «Розумного будинку» стає більш поширеною серед суспільства. Спочатку усі навіть вважали цю ідею неймовірною. А зараз ця концепція є майже у кожному домі[3].

У сучасної людини наявні численна кількість домашніх приладів та гаджетів. «Розумний будинок» дозволяє поєднувати їх між собою в єдину систему та контролювати з будь-якої точки світу[3].

Тобто розумний будинок – це система, яка забезпечує безпеку, ресурсозбереження та комфорт для всіх його користувачів. За рахунок цього в розумному будинку немає необхідності використовувати десятки різних пультів для кожного телевізора або кондиціонера, або постійно шукати вимикачі світла в темряві[3].

Зважаючи на популярність даної системи – вона буде інтегрована у велику кількість будинків вже за кілька десятиліть. У майбутньому, об'єднуючись у мережу, «Розумні будинки» будуть створювати «Розумні міста»[3].

1. 2 Етапи розробки системи «Розумний будинок»

Послідовність створення системи «Розумного будинку» і послідовне опрацювання етапів – попередня гарантія успішності системи. Перед початком його створення необхідно чітко зрозуміти та визначитися з завданням та функціями системи. Покрокове створення проєкту - це кропітка робота, яка пізніше дає дуже багато корисних функцій[4].

Створення «Розумного будинку» містить в собі не тільки підключення пристроїв до системи і написання коду, а й детальний аналіз проєкту,

співпрацю з замовником і пошук рішень для досягнення поставлених цілей проєкту. Процес розробки системи можна розділити на ключові етапи:

□ Визначення об'єму роботи. Перед початком створення системи потрібно визначити із замовником, які саме пристрої замовляти і у якому місці їх розташовувати. Також необхідно ознайомитись з конфігурацією будинку, розташування його поверхів та кімнат. При необхідності, визначитися з місцем для центрального пульта керування.

□ Створення чіткого технічного завдання. Технічне завдання - це основа, на яку покладається кожен фахівець, який бере участь в розробці. У випадку створення системи «Розумного будинку» визначаються декілька варіантів проєкту, які обговорюються із замовником за необхідності вносяться відповідні зміни і обирається найоптимальніший варіант. Тільки після цього проєкт переходить в стадію створення. На етапі формування і обговорення ТЗ також розробляється система, її розташування, кількість пристроїв та інтеграцій, їх послідовність розміщення і т.д.

□ Створення макету місць для розміщення пристроїв. У даному пункті нам потрібно підготуватися до встановлення. На даному етапі дуже важливо все прорахувати. Якщо це житловий будинок, то можна встановити датчики температури та вологості (в одній кімнаті або в декількох), датчики протікання води у ванній кімнаті, тощо. Також необхідно визначити місця де будуть розташовуватися дроти (кабелі), за допомогою яких відбувається з'єднання компонентів між собою, та електричні розетки для живлення пристроїв.

□ Встановлення та підключення пристроїв до НА. При встановленні пристроїв слід враховувати їх експлуатаційні характеристики.

□ Програмування. Напочатку може здатися, що ніякого коду у НА немає, але це не так. При розробці програми необхідно розуміти технічні характеристики пристрою, що використовується та його функціонал. В ньому

будуть відключені усі інтеграції, що приведе до непрацездатності всієї схеми. Щоб вирішити цю проблему потрібно знайти ту неправильну строку коду і виправити її. Але зазвичай система сама демонструє помилку в написанні або інші помилки. Під час створення програми обов'язково потрібно перевіряти конфігурацію системи для запобігання помилок в коді.

□ Тестування системи НА. Після усіх вищенаведених пунктів настає час перевірити працездатність всієї системи. З метою перевірки здійснюється тестове увімкнення «Розумного будинку» з поетапною перевіркою кожного її компонента у різних умовах експлуатації, зважаючи на його технічні характеристики.

– Демонстрація проєкта замовнику. Цей етап є завершальним. В ньому замовник приходить у приміщення, яке було оснащено системою НА та яка була налаштована на зручне користування. При наявності запитань у користувача здійснюється пояснення експлуатаційних особливостей системи, а також консультаційна підтримка у подальшому[4].

1.3 Аналіз існуючих програмних рішень та технологічних аналогів

При розробці системи «Розумний будинок» важливо проаналізувати готові програмні продукти, які вже є на ринку. Усі існуючі рішення можна розділити на кілька основних видів.

До першої групи належать готові комерційні екосистеми для звичайних споживачів. Вони створені для швидкого налаштування «з коробки» та дозволяють керувати пристроями різних виробників через один додаток. Наприклад, Google Home чудово працює на Android та підтримує голосове керування через Google Assistant. Платформа Apple HomeKit є закритою системою лише для користувачів iOS, але забезпечує високий рівень захисту даних та керується через додаток «Дім» або Siri. Платформа Amazon Alexa

підтримує найбільшу кількість розумних девайсів на ринку та ідеально підходить для управління через розумні колонки. Головний мінус цих систем — вони повністю залежать від інтернету та хмарних серверів компаній-виробників.

Другу групу складають локальні платформи з відкритим вихідним кодом (Open Source). Їх обирають для максимальної приватності, повної зміни коду під власні потреби та незалежності від хмари. Лідером серед таких рішень є Home Assistant. Він встановлюється на міні-ПК (наприклад, Raspberry Pi), підтримує тисячі пристроїв, дозволяє створювати гнучкі сценарії та може працювати взагалі без доступу до інтернету. Іншим відомим аналогом є платформа openHAB, яка написана на мові Java і має велику кількість готових плагінів. Також популярна система ioBroker, яка орієнтована на зручну візуалізацію даних та роботу з різними мережевими протоколами.

Будь-яке програмне забезпечення розумного будинку спирається на стандарти та протоколи зв'язку. Зараз активно розвивається новий універсальний протокол Matter, який дозволяє пристроям різних брендів безпечно спілкуватися між собою локально. Для бездротового зв'язку з низьким споживанням енергії використовуються протоколи Zigbee та Z-Wave, де кожен пристрій працює як ретранслятор сигналу. А для передачі даних від датчиків до сервера в розробках Інтернету речей (IoT) часто застосовують легкий протокол MQTT.

Якщо готові продукти не підходять для конкретних завдань, використовують індивідуальну розробку (Custom Development). Це включає написання власних програмних драйверів для нестандартного обладнання, створення вбудованого софту для мікроконтролерів (наприклад, ESP32 або Arduino) з нуля, а також розробку унікальних мобільних додатків або веб-панелей керування.

1.4 Визначення цілей та завдань системи

Формалізація місії, цілей та завдань проєкту є критично важливим етапом системного аналізу предметної області, оскільки вони визначають математичні та логічні орієнтири для всього процесу подальшої розробки програмного забезпечення.

Головною метою проєкту є розробка єдиної інтегрованої системи «Розумний будинок» на базі сучасних методів збору, обробки та передачі телеметричної інформації, що відповідає технічним критеріям надійності та масштабованості.

Для досягнення поставленої мети необхідно вирішити такі науково-технічні завдання:

- спроектувати та реалізувати центральний вузол обчислень для збору й агрегації даних від різнорідних сенсорів мікроклімату у режимі реального часу;
- розробити програмні модулі автоматизації для керування кліматичною технікою (кондиціонери, конвектори) на основі алгоритмів зворотного зв'язку;
- забезпечити високий рівень кібербезпеки та захисту персональних даних користувачів шляхом розгортання локально-орієнтованої архітектури та шифрування каналів зв'язку [5];
- розробити інтерактивний та інтуїтивно зрозумілий кросплатформний інтерфейс дистанційного керування та миттєвого сповіщення користувачів на базі Telegram Bot API [5].

Належний технічний супровід, тестування програмного коду та оптимізація алгоритмів взаємодії сутностей дозволяють створити конкурентоспроможну систему автоматизації, яка відповідає сучасним стандартам інженерії програмного забезпечення.

1.5 Висновок за розділ 1

У першому розділі було проведено загальний аналіз предметної області системи «Розумний будинок». Визначено актуальність використання сучасних технологій для підвищення комфорту, безпеки та енергоефективності житла.

На основі аналізу готових напрацювань було розглянуто популярні комерційні екосистеми, локальні платформи з відкритим кодом, базові протоколи бездротового зв'язку та варіанти індивідуального програмування. Також у розділі чітко сформульовано цілі та завдання проєкту.

Проведений аналіз підтвердив доцільність розробки власної автоматизованої системи та створив необхідну теоретичну базу для наступного етапу — обґрунтування і вибору конкретних програмно-апаратних засобів реалізації.

2 ВИБІР ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ

2.1 Популярні програмні засоби

При виборі програмних засобів для реалізації проєкту «система «Розумний будинок з інтеграцією чат-боту», необхідно враховувати різні аспекти, такі як функціональні вимоги проєкту, масштаб системи, якість пристроїв та різновид платформ для налаштування системи. Я вибирав між java, C++ та Python і відкритими платформами Home Assistant, openHAB, Domoticz, Tuuya Smart і вибрав Home Assistant. Я порівняв популярні програмні забезпечення у таблиці 2.1.

Таблиця 2.1 — Порівняння платформ

Критерій порівняння	Home Assistant (HA)	openHAB	Domoticz	Tuya Smart / Smart Life
Мова програмування ядра	Python	Java	C++	Java
Основна архітектура	Локальна (Local-first)	Локальна (Local-first)	Локальна (Local-first)	Хмарна (Cloud-based)
Формат конфігурації	YAML / Веб-інтерфейс	Специфічні файли конфігурації (.items, .rules)	Скрипти Lua / Blockly / Веб-інтерфейс	Мобільний додаток (No-code)
Споживання ресурсів	Помірне (рекомендовано від 1-2 ГБ RAM)	Високе (через віртуальну машину Java JVM)	Мінімальне (працює навіть на 256 МБ RAM)	Мінімальне на боці клієнта (все обчислюється в хмарі)

Продовження таблиці 2.1

Критерій порівняння	Home Assistant (НА)	openHAB	Domoticz	Tuya Smart / Smart Life
Кількість інтеграцій	Понад 2500+	Близько 400+	Обмежена (потребує ручного налаштування)	Обмежена лише пристроями партнерів Tuuya
Взаємодія з роботом (API)	Повна підтримка Webhooks, REST API, WebSockets	REST API	HTTP API / MQTT	Обмежений Tuya IoT Cloud API

Серед вище перерахованих платформ було вибрано саме Home Assistant. openHAB побудований на Java і використовує архітектуру OSGi. Це робить його корпоративно стабільним, але дуже «важким» для одноплатних комп'ютерів[5]. Home Assistant написаний на Python з використанням асинхронності, що забезпечує швидшу обробку подій (low latency) на Raspberry Pi 4[6].

Domoticz написаний на чистій мові C++, тому він майже не споживає оперативну пам'ять і процесорний час. Але він програє НА при підключенні нових технологій. При додаванні у Domoticz потрібно писати скрипти парсингу вручну на мові Lua[7].

Ось декілька популярних програм, які були використані:

□ Home Assistant: НА є найважливішою частиною цього проєкту. Вона об'єднує всі пристрої в єдину систему, чим значно спрощує життя користувача та допомагає контролювати всю систему.

□ Telegram: Один із найзручніших месенджерів, який широко використовується суспільством. Він надає дуже зручну та просту можливість створити чат-бота та налаштувати його як необхідно. Також Telegram простий в освоєнні, тому кожен зможе в ньому розібратися.

□ **Tuya Smart:** За допомогою данної програми можна підключити різноманітні пристрої від компанії Tuya. Ця програма потрібна для автоматичного додавання пристроїв до інтеграції системи.

Вибір конкретних програмних засобів залежить від специфічних потреб проєкту, вмінь розробників та вимог до системи. Ретельне обговорення з замовником та аналіз потрібних пристроїв допоможуть зробити правильний вибір програмних засобів для реалізації інформаційної системи «Розумного будинку».

2.2 Home Assistant:

Home Assistant (HA) - програмне забезпечення з відкритим вихідним кодом для домашньої автоматизації, підтримує пристрої різних виробників, забезпечує створення складних сценаріїв автоматизації з можливістю використання голосових помічників та візуалізацією за допомогою веб-інтерфейсу, а також програм для мобільних пристроїв[8]. Ось кілька характеристик, які описують Home Assistant:

□ **Гнучкість і масштабованість:** Home Assistant має гнучку архітектуру, яка дозволяє інтегрувати та керувати широким спектром пристроїв та сервісів, включаючи розумні світильники, датчики, звукові системи, термостати, медіацентри та багато іншого. Це дозволяє створювати індивідуальні рішення для управління розумним будинком відповідно до потреб користувача[9].

□ **Відкритий вихідний код:** Home Assistant є проєктом з відкритим вихідним кодом, що означає, що його код доступний для перегляду, модифікації та розповсюдження. Це дозволяє спільноті розробників створювати нові компоненти, інтеграцію та доповнення, розширюючи функціональність платформи[9].

□ Локальний контроль: Home Assistant надає можливість керування пристроями та виконання автоматизації безпосередньо на локальному рівні без необхідності звертання до хмарних сервісів. Це забезпечує підвищену безпеку та надійність роботи системи[9].

□ Автоматизація та сценарії: Home Assistant дозволяє створювати складні автоматизації та сценарії, засновані на різних умовах та діях. Ви можете налаштувати автоматичне увімкнення світла під час відкриття дверей, регулювання температури за певних погодних умов або включення медіацентру при приході додому[9].

□ Інтеграція з різними платформами: Home Assistant підтримує широкий спектр інтеграцій з популярними платформами та протоколами розумного будинку, такими як Zigbee, Z-Wave, Philips Hue, Google Assistant, Amazon Alexa, Apple HomeKit, Samsung SmartThings та багатьма іншими. Це забезпечує сумісність із великою кількістю пристроїв та сервісів[9].

□ Графічний інтерфейс користувача: Home Assistant надає веб-інтерфейс з інтуїтивно зрозумілим інтерфейсом користувача[9].

□ Гнучкі можливості автоматизації: Home Assistant пропонує широкий набір інструментів для створення складних автоматизацій. Ви можете використовувати автоматичні тригери, умови та дії для налаштування реакції системи на певні події. Це дозволяє створювати індивідуальні сценарії та підвищувати рівень автоматизації у вашому розумному будинку[9].

□ Розширюваність через плагіни та інтеграції: Home Assistant надає широкий вибір плагінів та інтеграцій, які дозволяють розширювати його функціональність. Ви можете додати нові пристрої, сервіси та функції до вашої системи розумного будинку, використовуючи доступні інтеграції або створюючи власні[9].

□ Доступність через різні платформи: Home Assistant доступний на різних платформах, включаючи Linux, Windows, MacOS, Docker, Raspberry Pi

та інші. Це дає вам гнучкість вибору відповідної платформи для вашого конкретного середовища[9].

□ Відкрита та активна спільнота: Home Assistant має активну та дружню спільноту користувачів та розробників. Існує форум, чати, репозиторії та інші ресурси, де ви можете отримати допомогу, обмінятися досвідом та знайти цікаві ідеї для використання платформи[9].

□ Безпека та приватність: Home Assistant приділяє особливу увагу безпеці та приватності даних. Він надає засоби для захисту системи та персональної інформації, а також дозволяє керувати доступом до функцій та пристроїв у вашому розумному будинку[9].

□ Підтримка різних мов та локалізація: Home Assistant підтримує безліч мов, що робить її доступною для широкої аудиторії користувачів по всьому світу. Завдяки активній спільноті, є переклади документації та інтерфейсу різними мовами, що полегшує використання платформи рідною мовою[9].

2.3 Telegram

Telegram — багатоплатформовий месенджер, що також надає опціональні наскрізні зашифровані чати (більш відомі як «секретні чати») та відеодзвінки, VoIP, обмін файлами та деякі інші функції. Він був запущений для iOS 14 серпня 2013 року та Android 20 жовтня 2013 року. Сервери Telegram розподілені по всьому світу з п'ятьма дата-центрами у різних частинах світу, а операційний центр базується в Дубаї, Об'єднані Арабські Емірати. Різні клієнтські програми доступні для настільних і мобільних платформ, зокрема офіційні програми для Android, iOS, Windows, macOS і Linux (хоча для реєстрації потрібен пристрій на базі iOS або Android і робочий номер телефону). Існує також два офіційних веб-додатки-двійники Telegram

— Web A та Web K, а також численні неофіційні клієнти, які використовують протокол Telegram. Офіційні компоненти Telegram мають відкритий вихідний код, за винятком сервера, який має закритий вихідний код і є власністю компанії[10].

Його засновник Павло Дуров сказав, що перша ідея застосунку з'явилася 2011 року, коли до нього приходили спецпризначенці РФ. Коли вони пішли, Дуров написав братові Миколі. Тоді ж він і усвідомив, що у нього немає безпечного способу комунікації з братом. Сервіс побудований на технології шифрування листування MTProto, розроблений братом Павла Миколою. Сам Telegram спочатку був експериментом компанії Digital Fortress (належить Павлові) з метою тестування MTProto на великих навантаженнях[10].

Для месенджера був створений протокол MTProto, що передбачає використання декількох протоколів шифрування. Під час авторизації і аутентифікації використовуються алгоритми RSA-2048, DH-2048 для шифрування, під час передачі повідомлень протоколу в мережу вони шифруються AES з ключем, відомим клієнту і серверу. Також застосовуються криптографічні хеш-алгоритми SHA-1 і MD5[10].

Безпека від перехоплення повідомлень, що пересилаються з боку сервера Telegram, забезпечується лише в режимі «секретних» чатів (Secret Chats), доступному з 8 жовтня 2013 року. Цей режим реалізує шифрування, при якому відправник і одержувач мають спільний лише для них ключ (end-to-end шифрування), із застосуванням алгоритму AES-256 у режимі IGE (англ. Infinite Garble Extension) для повідомлень, що пересилаються. На відміну від звичайного режиму, повідомлення в секретних чатах не розшифровуються сервером, історія листування зберігається лише на тих двох пристроях, на яких був створений чат[10].

У Telegram забезпечені наступні можливості:

□ Обмін файлами: Підтримується обмін фотографіями, відеозаписами та файлами будь-якого типу. Для фотографій доступна функція пошуку в інтернеті. Розмір файлу для надсилання обмежений 2 ГБ, а для користувачів із Telegram Premium — 4 ГБ. Програма використовує систему докачування файлів після обриву зв'язку[10].

□ Групові чати: У месенджері є можливість організовувати групові чати до 200 тисяч учасників. До 14 березня 2016 року максимальна кількість становила 200 учасників, після чого було запроваджено супергрупи, що вміщали до 5000 учасників. 17 червня 2018 року ліміт у супергрупі було збільшено до 100 тисяч, а пізніше — до 200 тисяч. Розробники вважають цю кількість достатньою й не планують збільшувати її[10].

□ Гілки: Режим гілок підтримують усі групи, що не прив'язані до каналу як чат для обговорення. Група може бути розділена на гілки, що присвячені певній темі обговорення. Гілки мають вигляд окремих підчатів усередині однієї групи — зі своїми повідомленнями, розділом із поширеними медіафайлами та налаштуваннями сповіщень. У всіх групах з режимом гілок існує розділ «Загальне». У ньому показуються повідомлення, що не належать до жодної гілки, зокрема ті повідомлення, що були надіслані до ввімкнення режиму гілок. Адміністратор може перейменувати цей розділ або керувати його видимістю. Є можливість відобразити всі повідомлення групи з гілками у вигляді єдиного чату, зокрема ті, що були надіслані до ввімкнення гілок. Решта користувачів можуть лише змінювати колір стандартної іконки гілки[10].

□ Дзвінки: У березні 2017 року в офіційних мобільних клієнтах Telegram з'явилася підтримка голосових викликів. Спочатку можливість зателефонувати була доступною користувачам із Західної Європи. Вона з'являлася і в тих, хто отримав виклик. Пізніше ця можливість стала доступною всім охочим. У травні 2017 року підтримка викликів з'явилася в

офіційних десктопних клієнтах. Виклики захищені шифруванням. Якість голосового виклику автоматично налаштовується залежно від якості зв'язку користувача. Так, під час користування мобільним інтернетом стиснення буде дещо вищим, ніж при з'єднанні через Wi-Fi. У 2020 році було додано відеодзвінки. Відеодзвінки, як і голосові дзвінки, підтримують peer-to-peer[10].

□ Боти: У Telegram працює платформа чатботів. Боти можуть виконувати різноманітні завдання, такі як пошук в інтернеті чи держреєстрах, покупки, платежі, розваги, модерація груп тощо. У спілкуванні беруть участь користувач Telegram та комп'ютерна програма від стороннього розробника. Користувач може взаємодіяти з ботом за допомогою елементів інтерфейсу месенджера: надсилання повідомлень, натискання на команди та кнопки, використання інлайн-режиму. Telegram надає три способи взаємодії користувача з ботом: приватний чат (класичний спосіб), група й так званий інлайн-режим:

□ Найпоширеніший спосіб — приватний чат. Бот може ініціювати діалог з користувачем лише у двох випадках: авторизація в сторонньому застосунку через Telegram і подання заявки на приєднання до чату.

□ Деякі боти можуть бути учасниками груп. Наприклад, у групах бот може підтримувати розмову, модерувати повідомлення або бути ведучим гри.

□ Інлайн-режим нагадує інтерфейс пошукової системи. Користувач уводить у поле для введення повідомлень запит, що починається з короткого імені бота. Далі користувач може вибрати й надіслати один з результатів[10].

Деякі боти можуть бути учасниками каналів. У каналах ботів застосовують здебільшого для запланованого розміщення повідомлень. З такими ботами користувач взаємодіє через приватний чат або вебінтерфейс. Бот взаємодіє з користувачами за допомогою Bot API. Керування ботами на платформі доступне в боті BotFather. Зокрема, там можна створити бота та

переглянути чи змінити присвоєний йому токен API. У вересні 2015 Дуров заявив про плани щодо монетизації й розміщення реклами в ботах. У листопаді 2020 року було презентовано можливість роботи Bot API на власному сервері. У 2022 році з'явилася можливість вебботів. Бот може показувати вікно з вебзастосунком, таким чином доволіно розширюючи стандартний інтерфейс месенджера. Також деякі боти отримали змогу бути доданими як розділ у меню вкладень і використовуватися в інших чатах подібно до онлайн-ботів[11].

2.4 TuYa Smart

Tuya Smart - міжнародна платформа, що фокусується на розробці рішень AI та IoT. Платформа TuYa Smart надає комплексні рішення, пов'язані з впровадженням домашньої автоматизації, починаючи від надання додатку та виділеної хмари, в якій працює система, до впровадження комунікаційних модулів та допомоги у їх впровадженні для масового виробництва[12].

Tuya Smart була заснована в 2014 році. Вже в 2015 році компанія отримала фінансування в розмірі \$15 млн, що дозволило компанії створити свій перший модуль Wi-Fi і запустити платформу IoT. У наступні роки компанія зробила подальші кроки, розпочавши стратегічну співпрацю з Google Home у 2017 році або приєднавшись до асоціації Bluetooth SIG у 2018. У 2019 році компанія відкрила кілька філій своєї компанії по всьому світу – в Індії, Японії, Німеччині та Колумбії. Джефф Іммельт, багаторічний президент General Electric (GE), також приєднався до команди TuYa, і компанія встановила ще одне партнерство, цього разу з Microsoft. У 2020 році компанія стала учасником Connectivity Standards Alliance, який об'єднує такі компанії, як Amazon, Google, Huawei, Signify та IKEA[12].

Принцип дії. Продукти Tuuya Smart спілкуються один з одним за допомогою бездротових протоколів передачі даних. Використовувані пристрої включають такі протоколи, як:

- Wi-Fi;
- ZigBee;
- Bluetooth Mesh;
- NB-IoT;
- GPRS;
- LTE.

Усі запити обробляються в обчислювальній хмарі Tuuya Cloud, що означає, що більшість пристроїв системи не потребують додаткової панелі управління будинком. Компанія має сервери по всьому світу, що дозволяє обробляти величезну кількість інформації без великих затримок. У 2018 році сервіс щодня обробляв 20 мільярдів запитів пристроїв і 6 мільйонів взаємодій AI[12].

Види пристроїв. Продукція Tuuya Smart створює дуже багату екосистему, так зване «Розумне життя». Завдяки правильному підбору продуктів це дозволяє повністю автоматизувати будинок з однієї програми. Серед доступних типів пристроїв ви можете знайти ряд пристроїв, керованих за допомогою безкоштовної спеціальної програми Тууа, до яких належать:

- Електрообладнання, таке як розетки та вимикачі;
- Освітлення, наприклад, лампи розжарювання та світлодіодні лампи;
- RTV та побутова техніка;
- Датчики та сигналізація;
- Дверні замки;
- Камери;

Завдяки створенню спеціальних модулів багато брендів створюють власні продукти на основі системи Tuуа, що робить її однією з найбільших Smart екосистем у світі. Дані за 2019 рік свідчать про використання понад 90 000 продуктів на основі компонентів екосистеми Tuуа, що дало близько 500 видів різних продуктів[12].

З початку своєї діяльності компанія Tuуа взяла на себе розвиток технології Smart. Тому протягом усього періоду роботи він активно підтримує поширення технологій, наприклад, організовуючи та беручи участь у присвячених цьому конференціям (AI+IoT Business Conference). У 2020 році Tuуа виступив співорганізатором конференції, присвяченої розумним рішенням на ринку Південно-Східної Азії, яка об'єднала понад 200 компаній, що працюють у сфері IoT[12].

У 2020 році компанія також відкрила відкриту платформу для хмарних розробників, яка на практиці має забезпечити подальший розвиток технологій AI та Iot через доступ до екосистеми Tuуа Smart[12].

2.5 Вибір апаратного забезпечення для реалізації інформаційної системи

Для нормального функціонування системи «Розумний будинок» потрібен пристрій, що з'єднає їх в єдиний ланцюг. Я вибирав між Raspberry Pi, Orange Pi, Banana Pi. При виборі були зважені їхні характеристики, які наведені в таблиці 2.2.

Таблиця 2.2 — Порівняння одноплатних комп'ютерів

Характеристика	Raspberry Pi 4	Orange Pi 5	Banana Pi M5
Процесор (CPU)	Broadcom BCM2711 (4 ядра, 1.5 ГГц)	Rockchip RK3588S (8 ядер, 2.4 ГГц)	Amlogic S905X3 (4 ядра, 2.0 ГГц)
Тип накопичувача	MicroSD / USB 3.0	MicroSD / Слот M.2 NVMe SSD	MicroSD / Вбудована eMMC 17 ГБ
Енергоспоживання	Низьке (~3-5 Вт)	Середнє (~4-9 Вт)	Низьке (~3-5 Вт)
Підтримка Home Assistant OS	Повна офіційна	Відсутня	Частково (сторонні образи)

У Orange Pi вища обчислювальна потужність за меншу вартість, але, в порівнянні з Raspberry, не має офіційної стабільної підтримки від HA ОС. А у разі виникнення помилок із драйверами чи портами GPIO Banana Pi знайти розв'язання проблеми в мережі складніше. Тому в проєкті був використаний Raspberry Pi 4[13].

Raspberry Pi — одноплатний комп'ютер, розроблений британським фондом Raspberry Pi Foundation. Головне призначення — сприяти вивченню базових комп'ютерних навичок школярами[14].

Щоб інформаційні дані від будинку потрапляли у систему потрібні датчики. Тому в цьому проєкті були взяті два датчики: температури та вологості і закритої двері. В подальшому можна використовувати і більшу кількість пристроїв для системи. У «Розумного будинку» різноманітне функціонування. Але цю систему можливо встановлювати не тільки вдома, а і в будь-якому приміщенні. Обмеження полягає лише в кількості можливих підключених пристроїв до роутера (він вміщує 120-130 пристроїв). Є різні способи підключення до системи, такі як: Bluetooth, ZigBee, Wi-fi. Ми використаємо Wi-fi через велику дальність роботи.

2.6 Raspberry Pi

Як правило в розумному будинку є центральний процесор – так звані «мізки» будинку. Цей процесор розпізнає конкретні ситуації, що відбуваються в будинку і реагує на них: керує поведінкою інших систем за допомогою заданих алгоритмів. У даному проекті він має назву Raspberry Pi 4.

Основний компонент Raspberry Pi - це друкована плата розміром приблизно з кредитну картку, на якій знаходяться процесор, пам'ять, різні порти вводу/виводу (наприклад, HDMI, USB, Ethernet, аудіо-вихід) та контакти для підключення інших пристроїв та датчиків. У комп'ютері задіяно 1,5-гігагерцовий процесор на архітектурі ARM; присутній роз'єм для навушників і слот для карти пам'яті[14]. Raspberry Pi працює під управлінням різних операційних систем, включаючи Raspbian (базується на Linux), Windows 10 IoT Core та інші. Ініціатором проекту Raspberry Pi є британський благодійний фонд Raspberry Pi Foundation[14].

Raspberry Pi може використовуватися в багатьох проектах, від створення домашнього медіацентру та ігрової консолі до автоматизації домашньої системи, робототехніки, інтернету речей та багато іншого. Завдяки своїй доступності та гнучкості, Raspberry Pi став популярним вибором для створення прототипів та DIY-проектів, а також використовується професіоналами у різних галузях[15].

2.7 Датчики

Існує безліч компаній, які займаються виробництвом датчиків для розумного будинку, зокрема: Tuuya, Samsung, Phillips, Xiaomi тощо.

Різновид датчиків та для будинку дуже багато. Наприклад, датчик температури та вологості заміряє параметри у приміщенні, де його

встановили. Система дає можливість користувачу за допомогою конвекторів або кондиціонерів змінити дані параметри дистанційно[16]. У цьому проєкті було використано датчики від Tuуа.

Датчики Tuуа є частиною цієї екосистеми та призначені для збору різних даних у розумному будинку. Вони можуть включати датчики руху, датчики відкриття дверей і вікон, датчики температури, вологості, датчики витоку води та інші.

Датчики Tuуа зазвичай працюють за бездротовими протоколами зв'язку, таким як Wi-Fi або Zigbee, і можуть бути легко інтегровані в розумний будинок через відповідні хаби або платформи управління. Вони зазвичай підтримують віддалене керування через мобільний додаток або голосові помічники, що дозволяє користувачам контролювати та моніторити стан свого будинку в режимі реального часу[17].

2.8 Wi-fi

Для підключення пристроїв до системи потрібна бездротова мережа. Є три основні бездротові мережі: Wi-fi, ZigBee та Bluetooth. У цьому проєкті буде використана мережа Wi-fi.

Wi-Fi (бездротова локальна мережа) – це технологія, яка дозволяє бездротове підключення пристроїв до Інтернету. Wi-Fi використовує радіочастотні сигнали для передачі даних між пристроями та бездротовим маршрутизатором або точкою доступу, які забезпечують підключення до дротової мережі Інтернет[18].

Wi-Fi дозволяє підключати різні пристрої, такі як комп'ютери, ноутбуки, смартфони, планшети, розумні телевізори, розумні домашні пристрої та інші до Інтернету без використання дротових з'єднань. Він також дозволяє створювати локальні бездротові мережі в будинках, офісах або

громадських місцях, де користувачі можуть обмінюватися даними та ресурсами.

2.9 Висновок за розділ 2

У другому розділі було повністю обґрунтовано вибір програмних та апаратних засобів для створення системи «Розумний будинок». На основі порівняльного аналізу існуючих аналогів для реалізації проєкту було обрано платформу Home Assistant та одноплатний комп'ютер Raspberry Pi 4, які разом забезпечують стабільну, швидку та автономну роботу системи в межах локальної мережі.

При розробці було чітко визначено інженерно-програмну структуру проєкту, де головні процеси обробки даних та керування подіями виконуються на мові Python, яка є основою обраної платформи автоматизації. Водночас усі сценарії автоматизації, алгоритми взаємодії та тригери, наведені у додатках роботи, написані за допомогою мови конфігурації YAML із використанням логічних шаблонів Jinja2.

Інтерактивний чат-бот розроблено на базі прикладного інтерфейсу Telegram Bot API, а його взаємодія із сервером автоматизації реалізована через технологію веб-хуків. Це дозволяє системі миттєво приймати команди від користувача, обробляти їх всередині платформи та надсилати зворотні сповіщення про стан приміщення.

Для надійного збору інформації про клімат та безпеку було обрано бездротові датчики температури, вологості та відкриття дверей екосистеми Tuuya Smart, які підключаються до головного сервера через домашню мережу Wi-Fi. Проведений аналіз повністю підтвердив правильність обраного технологічного стеку та створив необхідну базу для подальшого проєктування і практичної реалізації системи.

3 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ПРОГРАМНИХ КОМПОНЕНТІВ СИСТЕМИ

3.1 Об'єктно-модульна структура програмного забезпечення системи

Розробка програмного забезпечення інтегрованої системи автоматизації та моніторингу параметрів середовища базується на принципах модульності та мінімальної зв'язності програмних компонентів (low coupling). Це дозволяє забезпечити ізольовану відладку окремих підсистем, масштабованість архітектури та оптимізацію використання обчислювальних ресурсів як на рівні центрального сервера, так і на рівні периферійних пристроїв збору даних.

Логічна структура програмного забезпечення розподілена на два основні рівні: програмне забезпечення мікроконтролерного вузла збору даних (периферія) та архітектуру логічних менеджерів сервера керування (Home Assistant Core).

Логічну структуру програмних компонентів системи «Розумний будинок» наведено на рисунку 3.1.

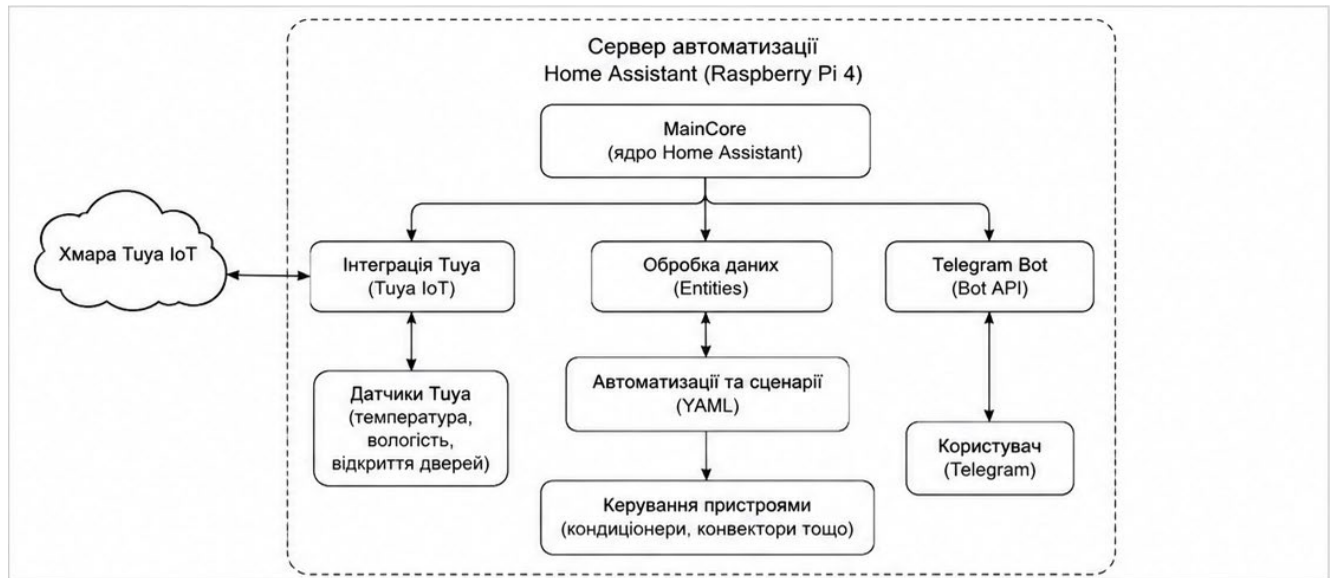


Рисунок 3.1 — Логічна структура програмних компонентів системи
«Розумний будинок»

- Отримання даних з датчиків TuYa. Датчики температури, вологості та відкриття дверей передають телеметричні дані по Wi-Fi до хмарної платформи TuYa IoT, після чого через інтеграцію TuYa IoT дані імпортуються у Home Assistant у вигляді сутностей (entities).
- Обробка даних. Модуль обробки даних отримує значення сутностей у реальному часі та формує події системи, які використовуються в сценаріях автоматизації.
- Автоматизації. Сценарії, правила та умови керування описуються у конфігураційних файлах YAML, де визначаються реакції системи на події та стани датчиків.
- Взаємодія з користувачем. Через інтеграцію Telegram Bot API чат-бот забезпечує передачу команд користувача до системи та надсилання повідомлень про стан приміщення і тривожні події.
- Керування пристроями. На основі сценаріїв система керує виконавчими пристроями (кондиціонерами, конвекторами тощо) для підтримки заданих параметрів мікроклімату.

На рівні центрального сервера автоматизації розгорнуто подієво-орієнтовану машину станів, архітектуру логічних компонентів якої наведено на рисунку 3.2.



Рисунок 3.2 — Діаграма логічної структури програмного забезпечення сервера

Представлена на рисунку 3.2 діаграма логічної структури відображає архітектурні рівні обробки інформаційних потоків та показує, як саме взаємодіють програмні компоненти сервера. На програмному рівні в межах платформи Home Assistant ця логіка реалізується за допомогою декількох взаємопов'язаних системних модулів та інтеграцій.

Взаємодія всередині сервера виконується через асинхронну шину подій ядра системи (Event Bus), яка є головним диспетчером обміну повідомленнями

у реальному часі. Програмний компонент Home Assistant Core ініціалізує та оновлює стани об'єктів у єдиній внутрішній машині станів (State Machine), де кожен фізичний пристрій представлений у вигляді програмної сутності (Entity). Прийом даних від бездротової периферії здійснюється за допомогою інтегрованого програмного модуля Tuuya Integration. Цей модуль підтримує постійне WebSocket-з'єднання з хмарою або локальним пристроєм, парсить вхідні пакети даних і автоматично передає оновлені значення температури та вологості до машини станів.

Оброблена телеметрія передається до вбудованого двигуна автоматизацій Automation Manager, який відповідає за інтерпретацію та виконання сценаріїв керування, описаних у конфігураційних файлах YAML. Зворотний зв'язок із користувачем та обробка зовнішніх запитів реалізовані за допомогою компонента Telegram Bot Integration. Цей модуль конфігурує веб-хук (Webhook) для прийому вхідних HTTP POST-запитів у форматі JSON від серверів Telegram Bot API, розпізнає унікальні ідентифікатори команд та передає їх на виконання до внутрішньої шини подій, забезпечуючи миттєву інтерактивну взаємодію з інтерфейсом чат-бота.

3.2 Обґрунтування стеку системних бібліотек та фреймворків

Для забезпечення стабільної взаємодії між апаратними компонентами, програмними сервісами та зовнішніми прикладними інтерфейсами (API) було застосовано стек спеціалізованих бібліотек. Перелік, призначення та версії використаних програмних засобів наведено в таблиці 3.1.

Таблиця 3.1 — Системні компоненти та бібліотеки, що використовувалися при реалізації системи

Назва компонента / бібліотеки	Призначення в інфраструктурі системи	Версія
Home Assistant Core	Основне середовище виконання асинхронної машини станів	2026.5.4
Назва компонента / бібліотеки	Призначення в інфраструктурі системи	Версія
aiohttp	Реалізація асинхронного HTTP-сервера для обробки Webhooks від Telegram	3.10.1
Tuya IoT Integration	Трансляція та парсинг даних з хмарного Tuya IoT Cloud API через WebSockets	5.1.0
Jinja2 Template Engine	Динамічний рендеринг повідомлень та обробка інлайн логіки в реальному часі	3.1.4
PyYAML	Синтаксичний аналіз, десеріалізація та валідація конфігураційних файлів YAML	6.0.1
SQLite Driver (sqlite3)	Організація реляційного зберігання та логування історії зміни станів датчиків	3.45.0

3.3 Розробка прикладного протоколу та структури пакетів даних

Для оптимізації мережевого трафіку, зменшення навантаження на бездротові канали зв'язку та зниження енергоспоживання периферійних пристроїв під час автономної роботи, було розроблено компактну побайтову структуру пакета прикладного рівня для передачі телеметрії. Побайтовий формат корисного навантаження інформаційного пакета наведено в таблиці 3.2.

Таблиця 3.2 — Побайтова структура пакета телеметрії прикладного рівня

Зсув (Байт)	Назва поля пакета	Тип даних	Опис та фізичний зміст поля
0	MsgType	uint8_t	Тип повідомлення (0x01 – дані вимірювань, 0x02 – системний статус)
1-2	Humidity	uint16_t	Показник відносної вологості (масштабовано: значення $\times 100$)
3-4	Temperature	int16_t	Показник температури середовища (масштабовано: значення $\times 100$)
5-8	Timestamp	uint32_t	Часова мітка фіксації події в системі (Unix Timestamp)

Продовження таблиці 3.1

Зсув (Байт)	Назва поля пакета	Тип даних	Опис та фізичний зміст поля
9	BatteryLevel	uint8_t	Рівень заряду автономного джерела живлення пристрою (0 – 100%)
10	SysFlags	uint8_t	Системні бітові прапори стану (Bit 0 – зв'язок, Bit 1 – помилка сенсора)

3.4 Організація інформаційного обміну за допомогою технології Webhooks

Взаємодія компонентів системи підпорядкована подієво-орієнтованому алгоритму обробки запитів користувача. Маршрутизацію інформаційних потоків між інтерфейсним рівнем, сервером автоматизації та кінцевими датчиками приміщення наведено на рисунку 3.3.



Рисунок 3.3 — Схема інформаційних потоків та автоматизації системи

Обмін повідомленнями базується на асинхронній технології Webhooks. При натисканні користувачем інлайн-кнопки в інтерфейсі чат-бота, хмара Telegram Bot API генерує структурований JSON-пакет і надсилає його за допомогою HTTP POST-запиту на шлюз сервера Home Assistant.

Компонент `CommunicationManager` перехоплює цей пакет, верифікує права доступу клієнта за білим списком `allowed_chats` та ініціює подію внутрішньої шини `telegram_callback`. Запуск відповідної автоматизації активує опитування поточного зрізу станів потрібних сутностей, виконує динамічний рендеринг відповіді через текстовий рушій шаблонізації `Jinja2` та відправляє сформований Markdown-текст назад користувачу.

4 НАЛАШТУВАННЯ І ТЕСТУВАННЯ СИСТЕМИ

4.1 Розробка та налаштування системи

Розробка та налаштування є однією з найбільш важливих складових процесу створення проєкту «Розумний будинок». Цей процес передбачає підключення датчиків до системи та інтегрування чат-бота за допомогою написання коду у ядрі НА.

Під час розробки проєкту була врахована можливість розширення пристроїв системи та адаптація під будь-яке приміщення. Наявна можливість використання даної системи, наприклад, як для однієї окремої квартири, так і для будинку в цілому або готелю. Також не є винятком і встановлення НА у державних установах або на підприємствах. У подальшому можливе додавання нового обладнання, зокрема, датчиків, пристроїв, побутової техніки тощо.

Управління повинно бути зрозумілим та легким для користування, а також містити достатній потенціал та функціональність, щоб задовольнити потреби користувачів.

Після підключення датчиків до системи потрібно налаштувати інтеграцію чат-бота для зручності керування. Це включає створення чат-бота у Telegram, написання коду у ядрі НА та налаштування автоматизацій для різних пристроїв.

Окрім цього, під час підключення датчиків до системи необхідно забезпечувати їхнє джерело живлення та стабільність роботи. Це включає в себе заміну елементів живлення (батарей) або заживлення їх від зовнішньої електричної мережі. Для стабільності роботи необхідно обрати з широкого асортименту товарів для розумного будинку придатні для конкретного проєкту за різними визначеними параметрами та використовувати продукцію відомих виробників з потрібними характеристиками товарів.

З метою ефективного налаштування системи «Розумний будинок», необхідно врахувати специфіку використання житлового приміщення (чи постійно в ньому мешкають люди; чи приміщення використовується зрідка тощо) та потреб користувача.

«Розумний будинок» - це система, яка дозволяє відстежувати показники у визначеному приміщенні та керувати деякими процесами в ньому.

Також, важлива правильність написання коду в ядрі, конфігурація системи та розуміння що і як саме працює.

Загалом, розробка та налаштування системи «Розумний будинок» з інтегрування чат-бота має бути спрямована на спрощення різних побутових процесів та зручність користування.

4.2 Налаштування системи

У ході налаштування системи було проведено наступні дії для розробленої системи «Розумний будинок»:

□ Встановлення НА на Raspberry Pi: для встановлення системи потрібна карта пам'яті приблизно на 16 Gb та CardReader для перенесення файлів з ПК. Вікно програми для встановлення НА на Raspberry Pi. (Рисунок 4.1)



Рисунок 4.1 – Вікно програми встановлення

При встановленні системи карту пам'яті буде відформатовано. Перед цим потрібно назвати карту пам'яті «CONFIG», як зазначено на рисунку 4.2

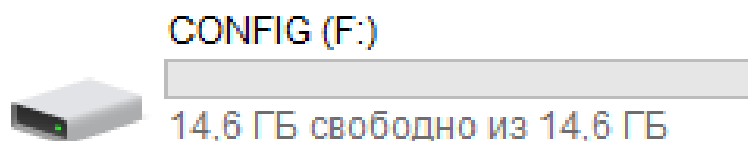
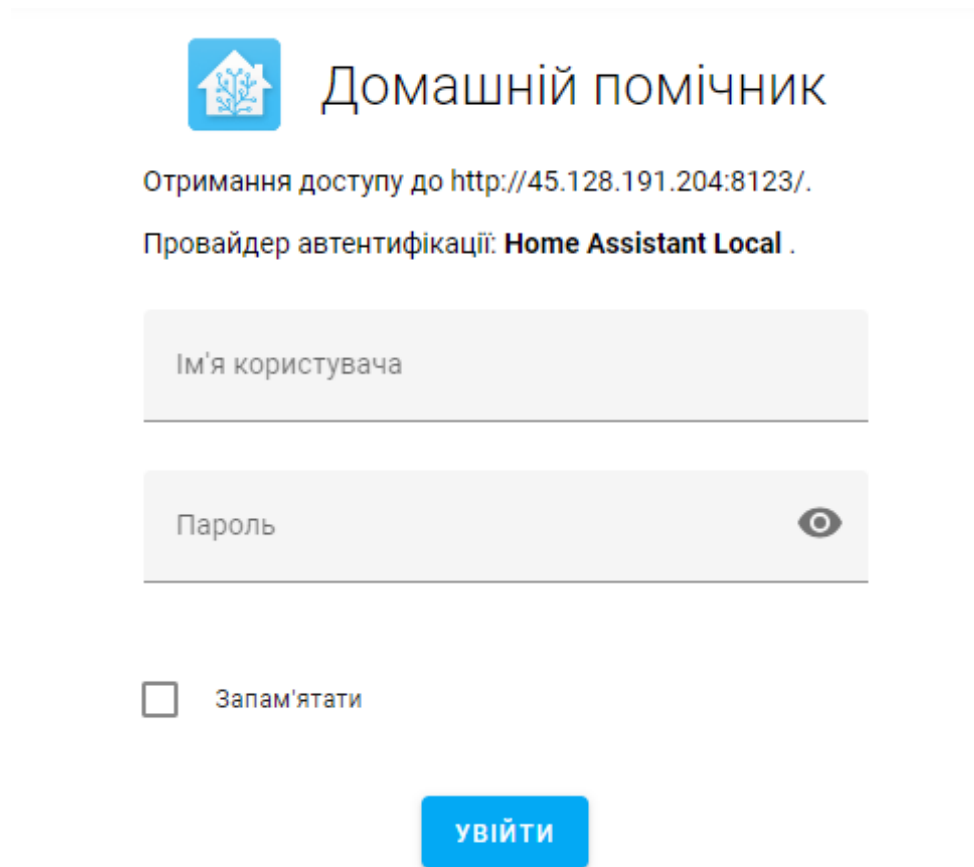


Рисунок 4.2 – Назва карти пам'яті

Після цих процесів потрібно витягнути CardReader та від'єднати карту пам'яті. Вставляємо карту пам'яті у Raspberry Pi. Потрібно почекати декілька секунд для того, щоб на сервер завантажилася система НА.

□ Створення акаунту в НА: при реєстрації потрібно вказати своє місце проживання та потрібно знати IP адресу Raspberry. Для поліпшення користування краще замовити у провайдера білу адресу. Ця функція дозволяє зайти в систему з іншого пристрою. Але при вході все одно необхідно знати білий IP, логін та пароль. Вікно реєстрації у НА. (Рисунок 4.3)



The image shows a web interface for the 'Домашній помічник' (Home Assistant) authentication provider. At the top, there is a blue icon of a house with a plant inside, followed by the title 'Домашній помічник'. Below the title, it says 'Отримання доступу до http://45.128.191.204:8123/' and 'Провайдер автентифікації: Home Assistant Local'. There are two input fields: 'Ім'я користувача' (Username) and 'Пароль' (Password). The password field has an eye icon to toggle visibility. Below the password field is a checkbox labeled 'Запам'ятати' (Remember). At the bottom, there is a blue button labeled 'УВІЙТИ' (Log In).

Рисунок 4.3 – Вікно реєстрації

Спочатку потрібно визначити IP адресу Raspberry Pi. З метою спрощення використання НА потрібно зарезервувати визначений IP. Це необхідно для того, щоб не дізнаватися IP системи при кожному вході. При реєстрації акаунту потрібно ввести ім'я користувача та пароль. Після цього потрібно вказати місце, де ти мешкаєш, за допомогою мапи. (Рисунок 4.4)

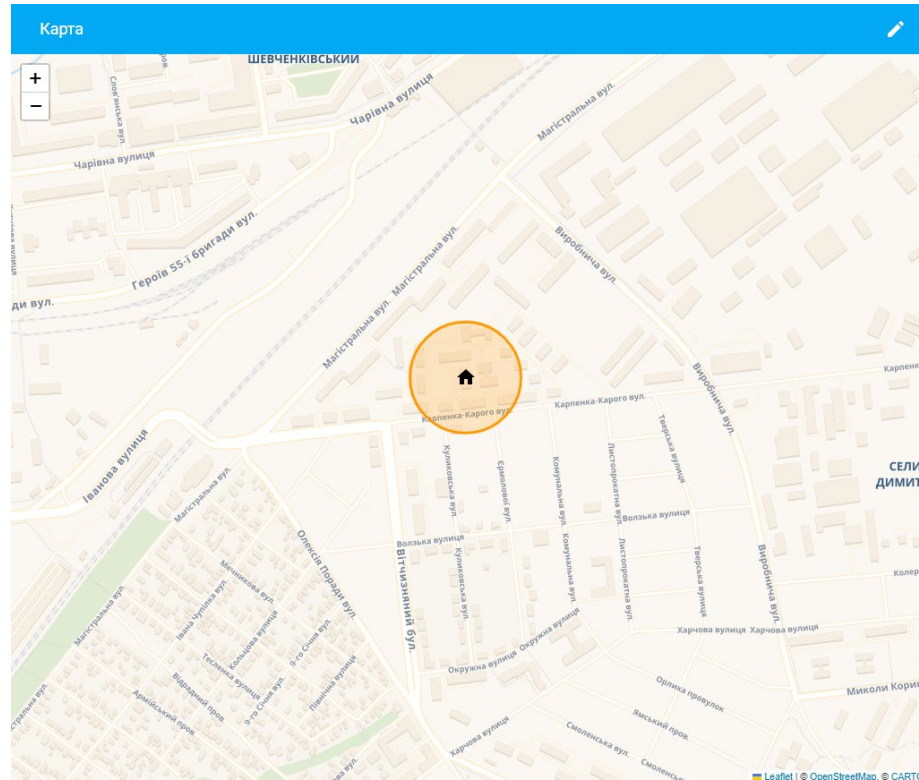


Рисунок 4.4 – Мапа

Визначення місцезнаходження необхідно для точної та стабільної роботи системи.

□ Підключення датчиків до системи: при підключенні датчиків до НА потрібно враховувати виробника та спосіб підключення до системи. Меню інтеграцій, в якому спочатку система демонструє те, що вона сама знайшла та можливість додати нові. (Рисунок 4.5)

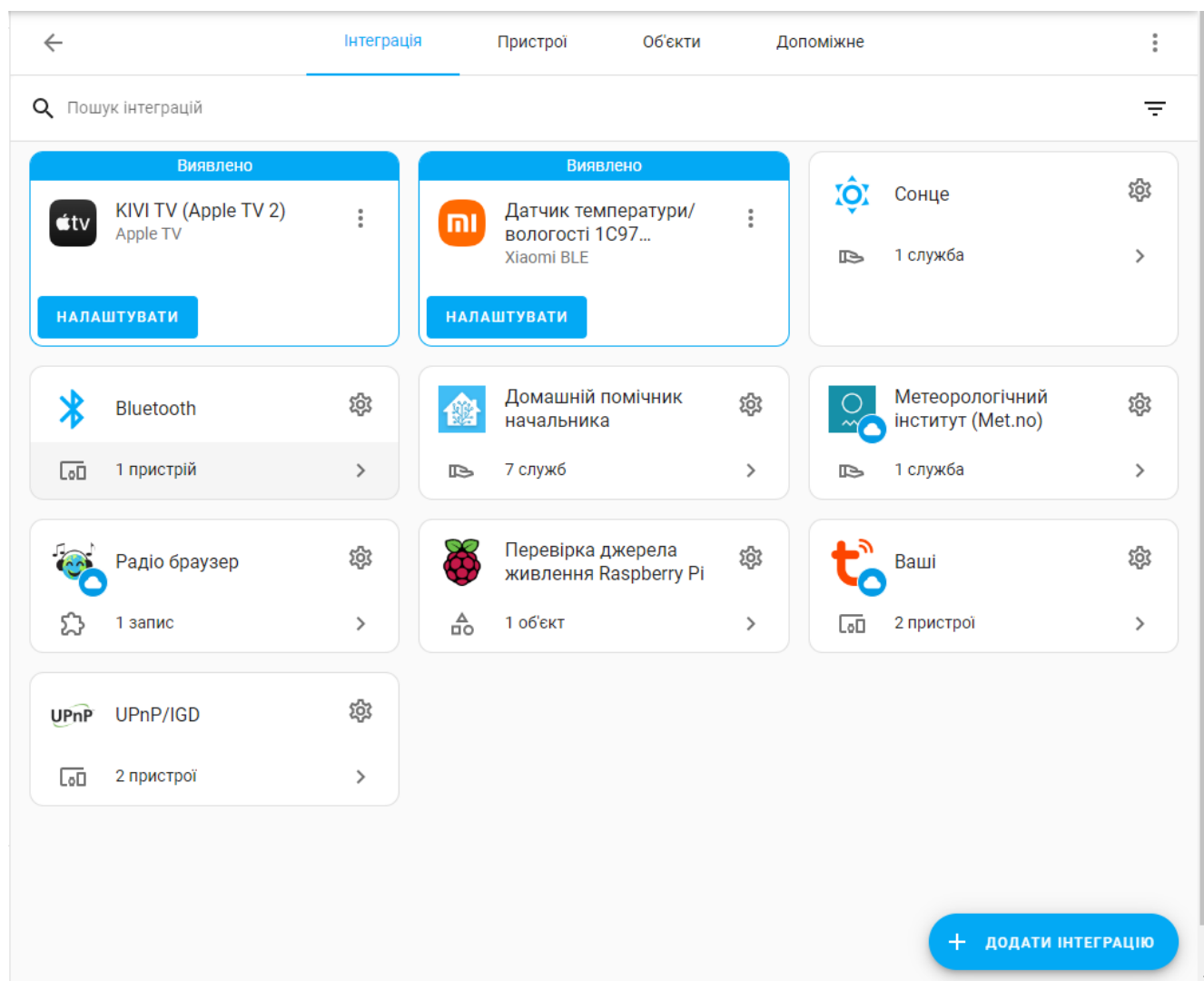


Рисунок 4.5 – Меню інтеграцій

– При додаванні з'явиться вікно списку інтеграцій визначених виробників. (Рисунок 4.6)

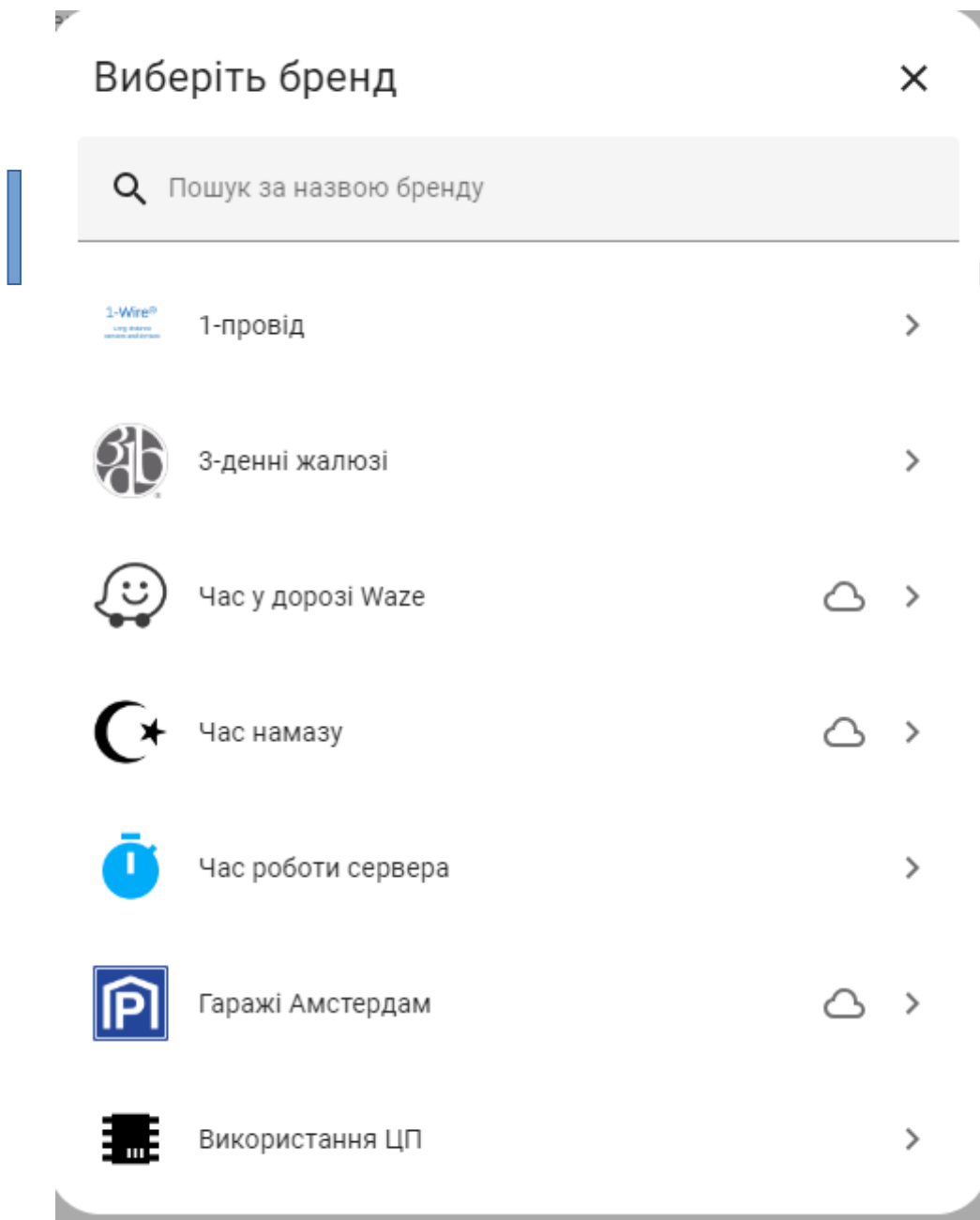


Рисунок 4.6 – Вибір інтеграції

4.3 Підключення датчиків до TuYa IoT

Для проєкту були вибрані датчики від компанії TuYa. Було обрано такі датчики: датчик температури та вологості і датчик закритої двері. (Рисунок 4.8 та 4.9)



Рисунок 4.8 – Датчик температури та вологості



Рисунок 4.9 – Датчик закритої двері

Перед початком підключення датчиків до системи – їх потрібно підключити до мобільного додатку «Smart Life». (Рисунок 4.10)

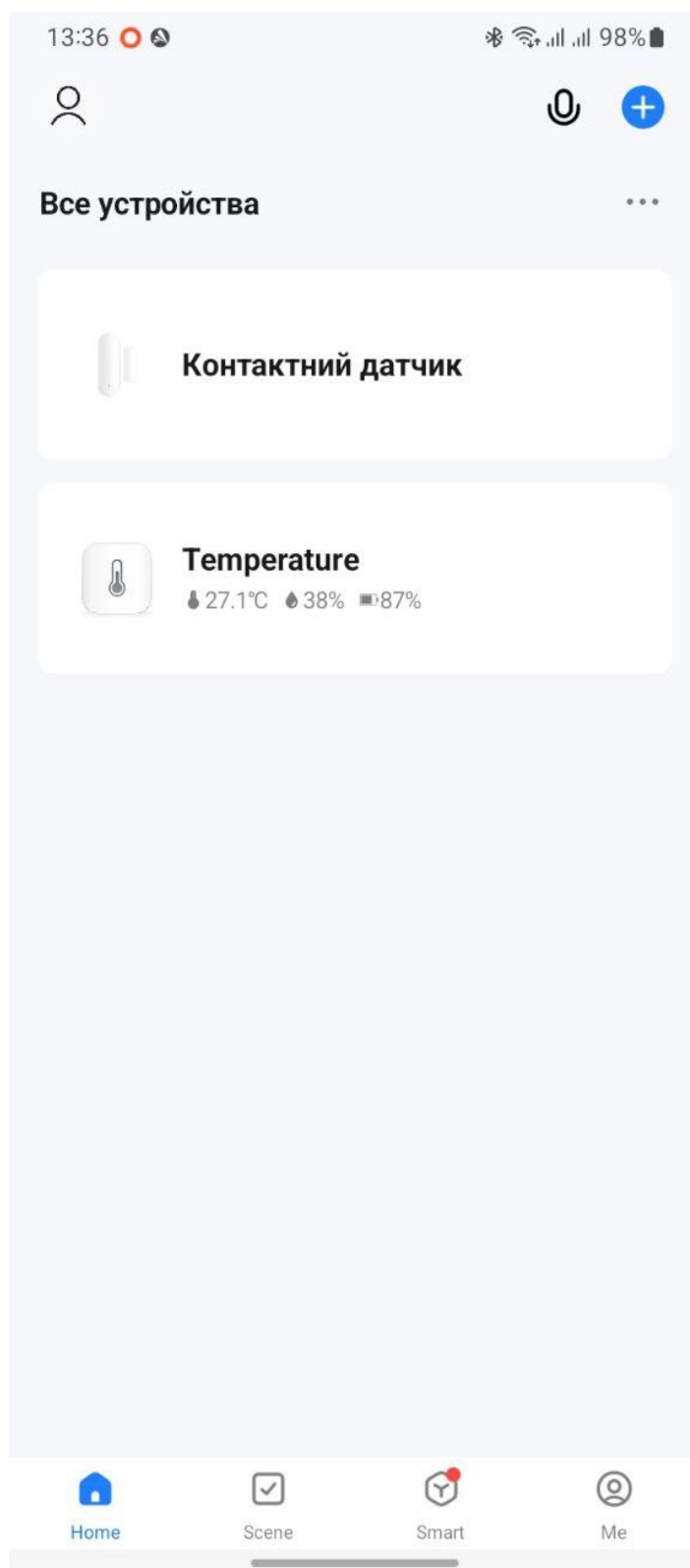


Рисунок 4.10 – «Smart Life»

У коробці до датчиків є інструкція, в якій описано як їх додати. Здійснюємо їх додавання, використовуючи інструкцію.

Тепер, перевіривши працездатність, переходимо на офіційний сайт TuYa. Його головна сторінка виглядає наступним чином. (Рисунок 4.11)

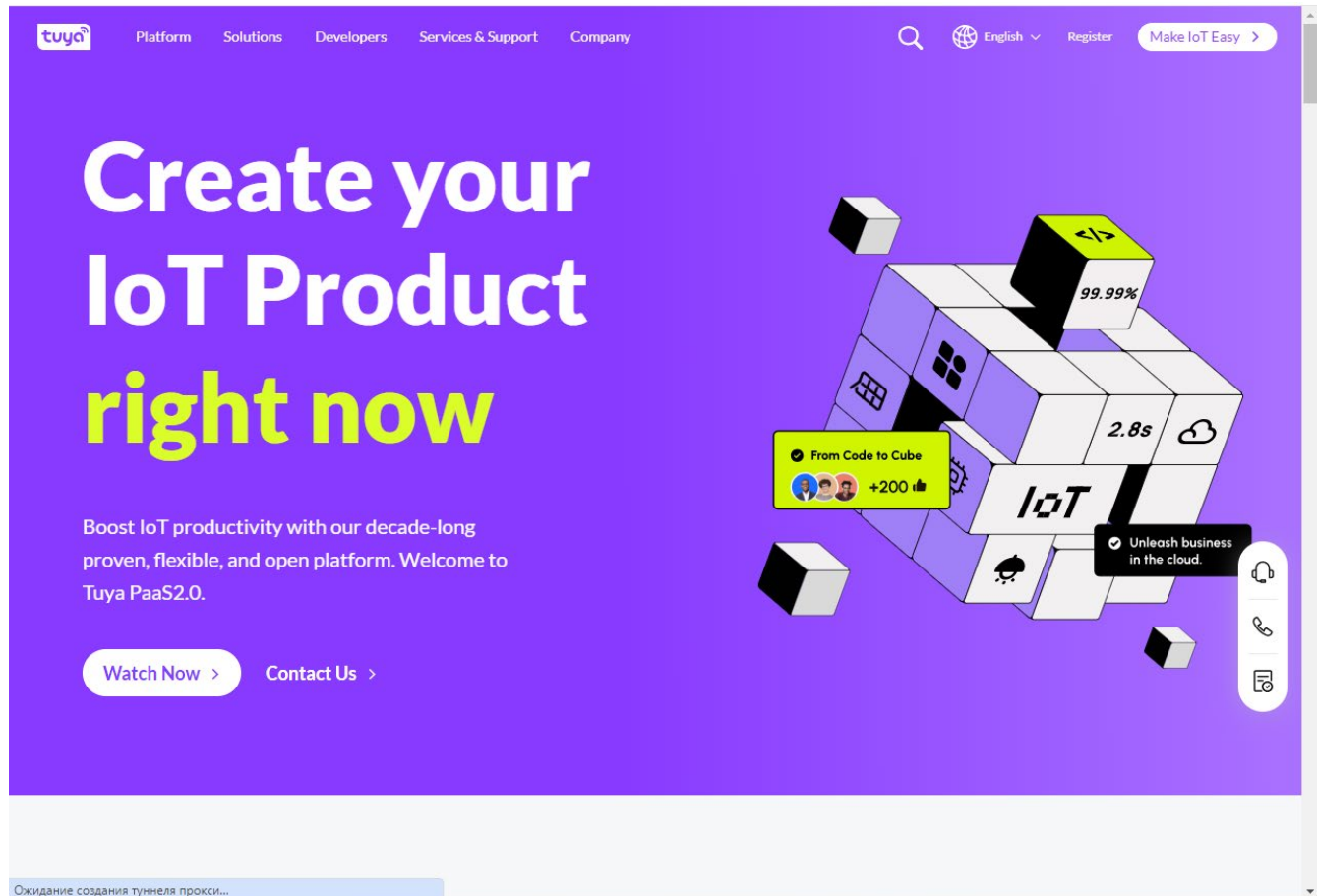


Рисунок 4.11 – Головна сторінка TuYa

На сайті TuYa натискаємо на “Make IoT Easy” та переходимо на TuYa IoT. Перед цим нам потрібно зареєструватися для роботи з сайтом. Реєстраційне вікно. (Рисунок 4.12)

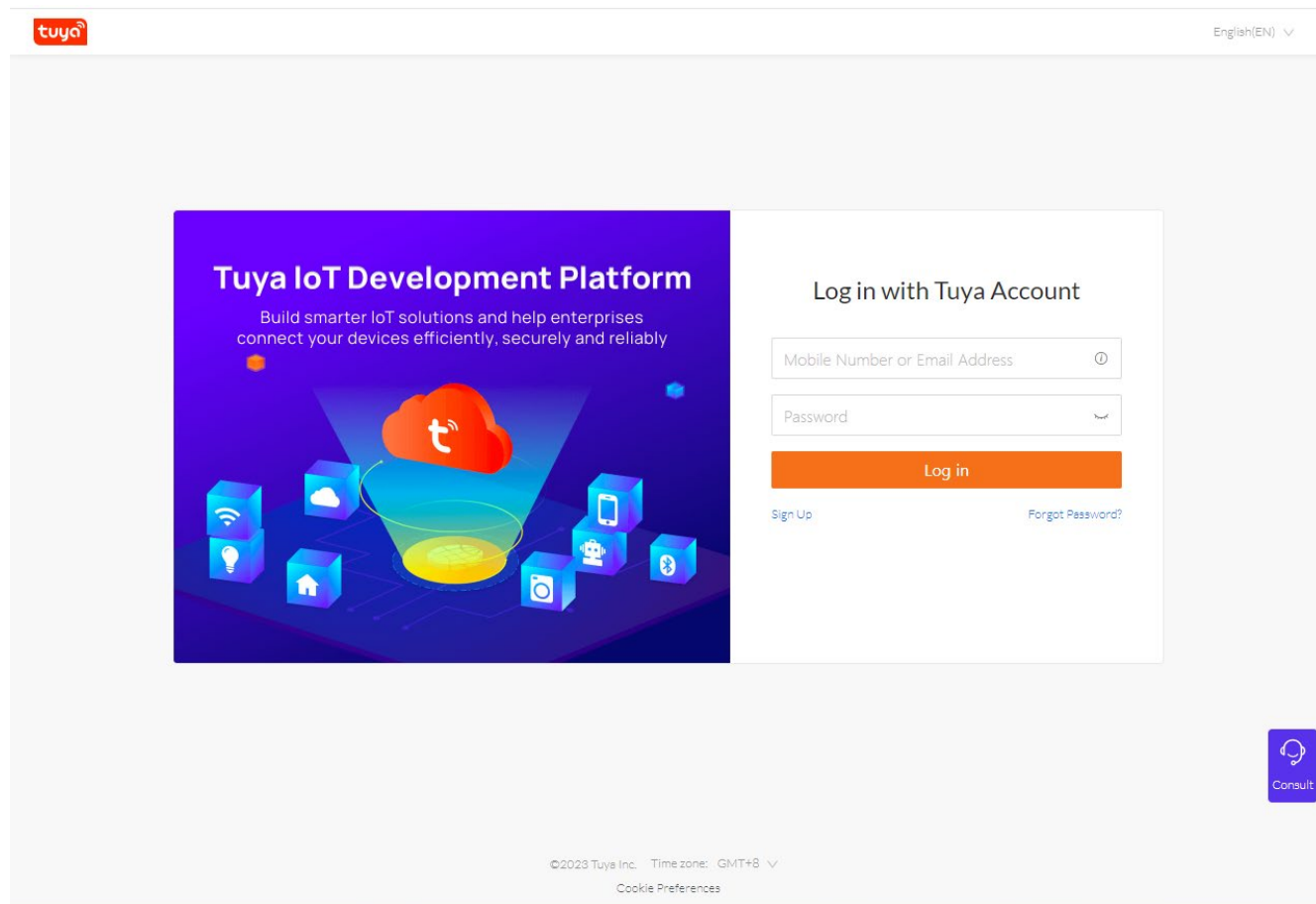


Рисунок 4.12 – Реєстраційне вікно

Після реєстрації потрапляємо на головну сторінку «Tuya IoT». Далі натискаємо «Cloud». Створюємо проєкт за допомогою кнопки «Create Cloud». Ми потрапляємо на меню додавання датчиків. (Рисунок 4.13)

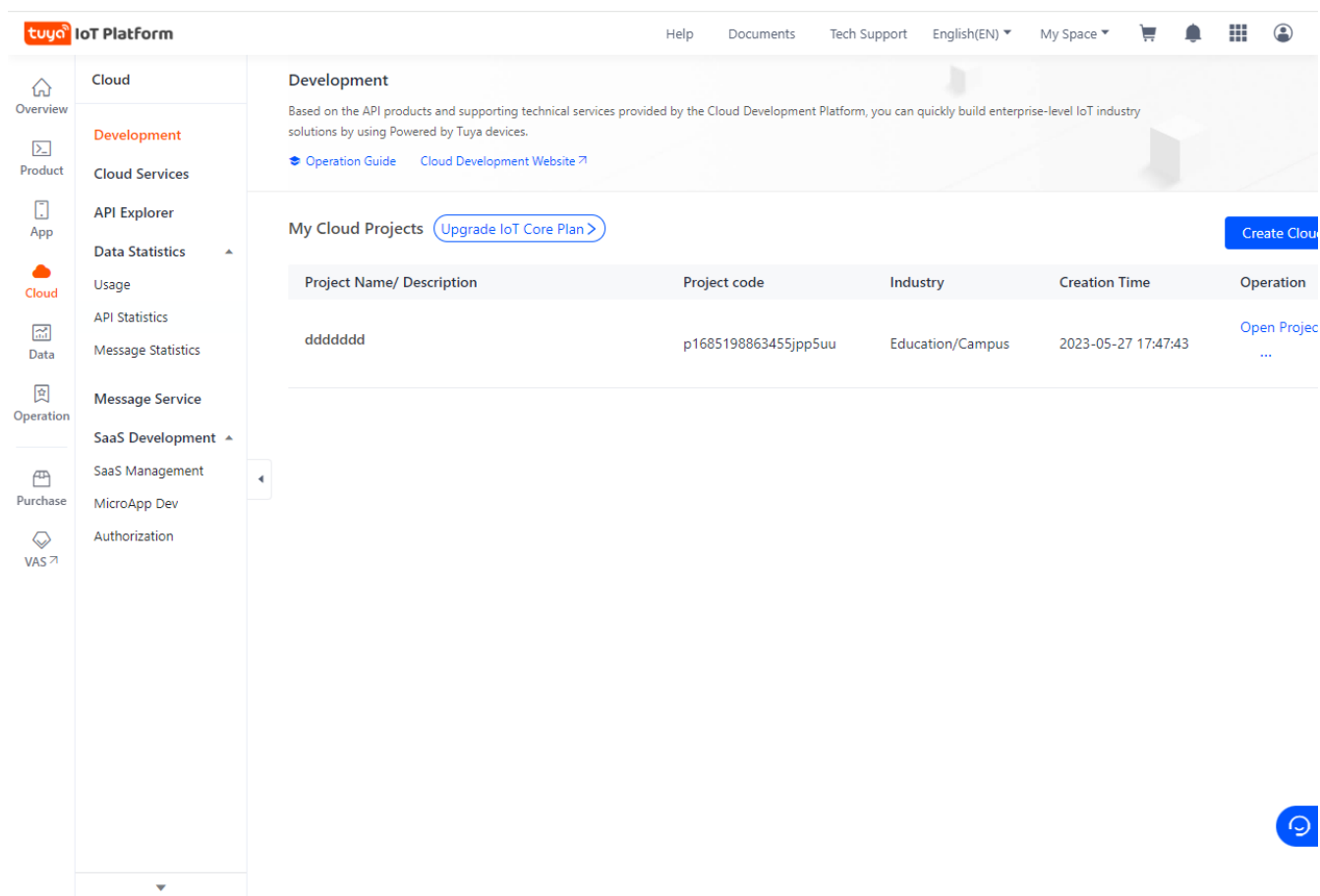
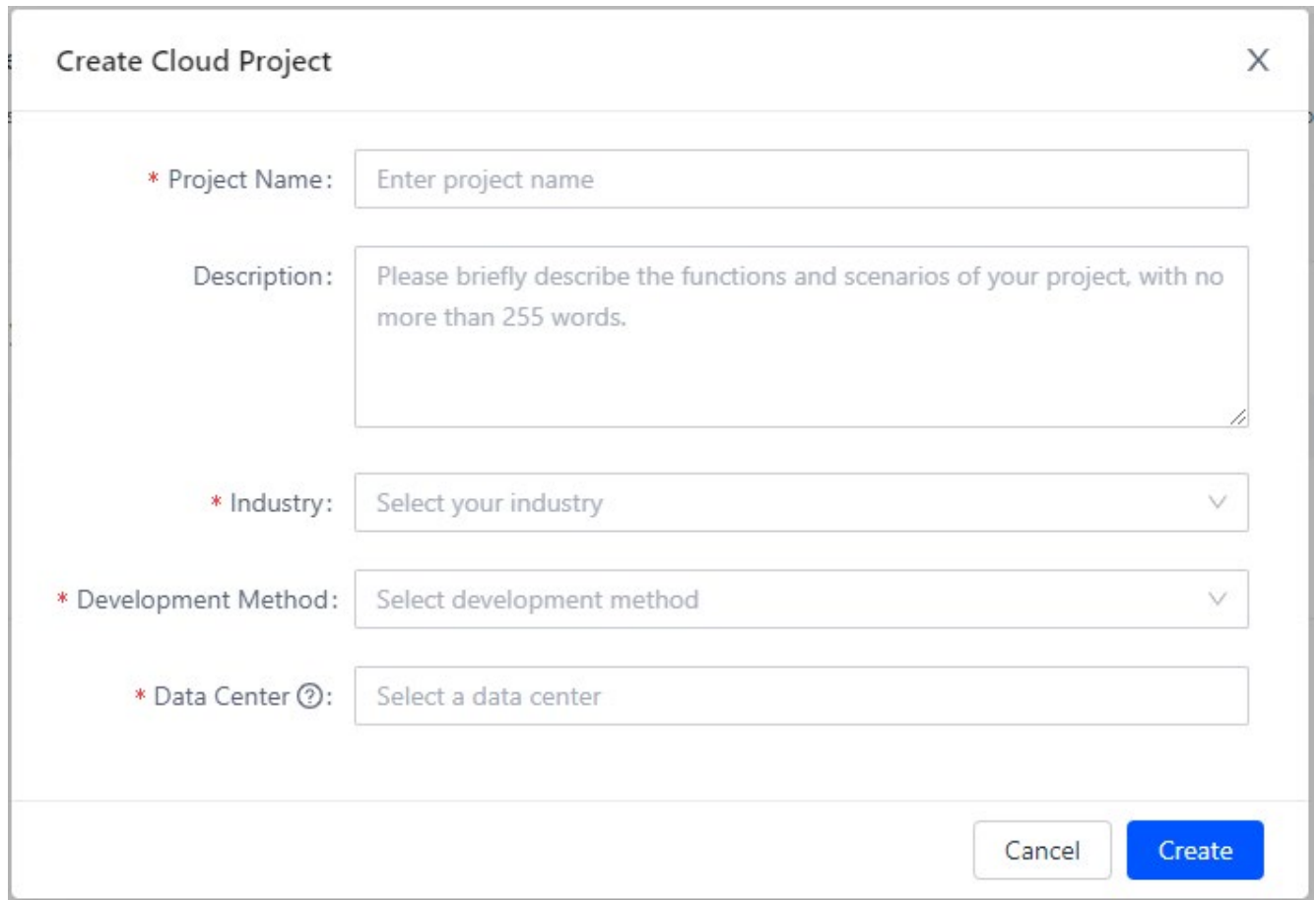


Рисунок 4.13 – Меню «Cloud»

За допомогою цього проєкту буде простіше додавати пристрої від Туя, бо він буде одразу синхронізувати нові пристрої, які будуть підключатися до додатку «Smart Life».

Після натиснення кнопки створення нового проєкту відкриється меню заповнення інформації. (Рисунок 4.14)



Create Cloud Project X

* Project Name:

Description:

* Industry:

* Development Method:

* Data Center ? :

Cancel Create

Рисунок 4.14 – Меню створення проєкту

У цьому вікні вказуються такі значення: ім'я проєкту, опис, програма, в якій будуть спочатку підключатися пристрої (в нашому випадку це «Smart Life»), метод передачі («Smart home» або «Custom») та місце знаходження системи (наприклад, Central Europe Data Center).

Після створення проєкту потрібно його відкрити, натиснувши «Open Project». З'являється меню нашого проєкту. (Рисунок 4.15)

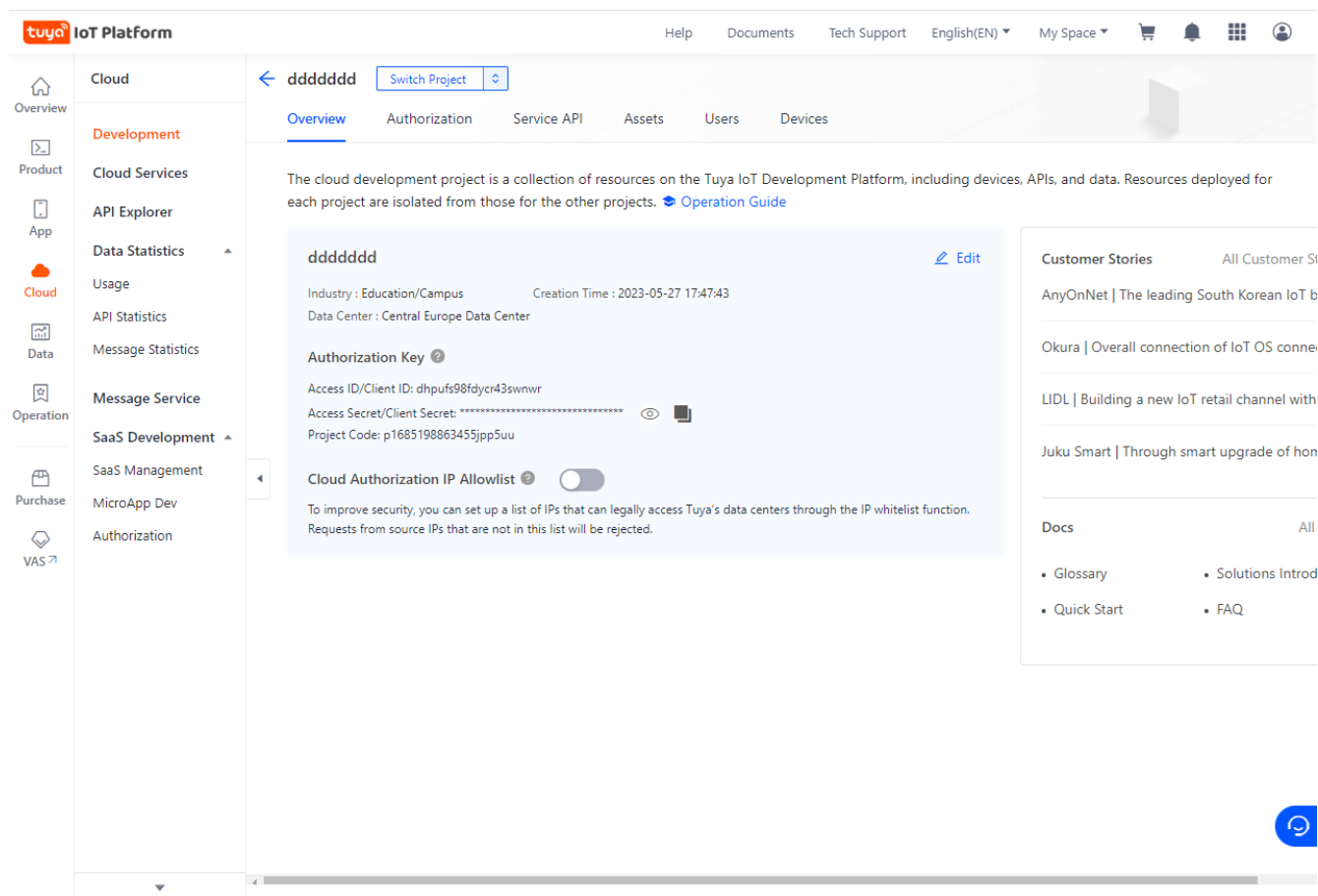


Рисунок 4.15 – Меню проєкту

В ньому в самому початку написані дані, які знадобляться при підключенні інтеграції НА, а саме Client ID та Client Secret.

Для додання пристроїв до проєкту потрібно перейти в меню «Devices». У цьому вікні потрібно перейти в «Link TuYa App Account». (Рисунок 4.16)

tuya IoT Platform Help Documents Tech Support English(EN) My Space Central Europe Data Center

Overview Authorization Service API Assets Users **Devices**

You can add devices to a specified asset under a project by using the IoT Device Management app, and can also link device resources of other apps. [Operation Guide](#) [View Device Linking Methods](#)

All Devices Link My App Link My WeChat Mini Program **Link Tuya App Account** Link SaaS

You can link devices to this project by using accounts of Smart Life App. Already added 1 accounts and linked 2 devices. [Add App Account](#)

App Account	UID	App Name	Devices	Linked Projects	Device Linking Method	Linked Devices	Operation
za.llerkov.nikron@gmail.com	eu1684955117739rqybX	Smart Life	2	ddddddd	Automatic Link	2	Manage Devices Unlink

Total Items: 1 < 1 > 10 Items/Page

Рисунок 4.16 – Меню «Link Tuya App Account»

Натискаємо кнопку «Add App Account» для синхронізації додатку з проєкту. Відкривається QR-код, який необхідно просканувати за допомогою «Smart Life» (або іншого додатку). (Рисунок 4.17)

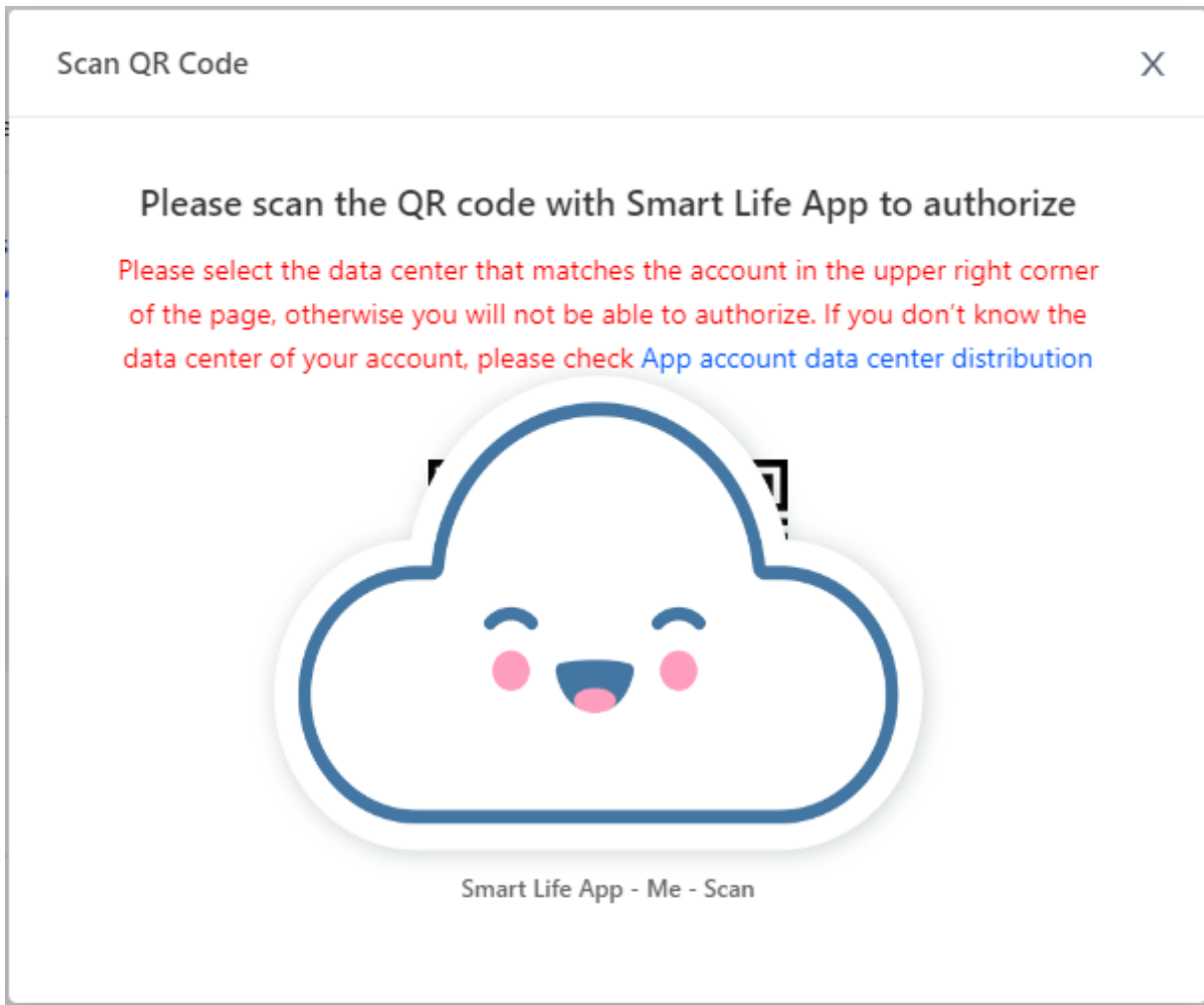


Рисунок 4.17 – QR-код (замальований)

Тепер всі пристрої, які будуть підключатися до додатку, автоматично підключаться до проєкту Tuuya IoT.

4.4 Підключення датчиків до НА

Тепер потрібно підключити проєкт Tuuya IoT до системи Home Assistant. Для цього заходимо в налаштування та вибираємо пункт «Пристрої та служби». (Рисунок 4.18)

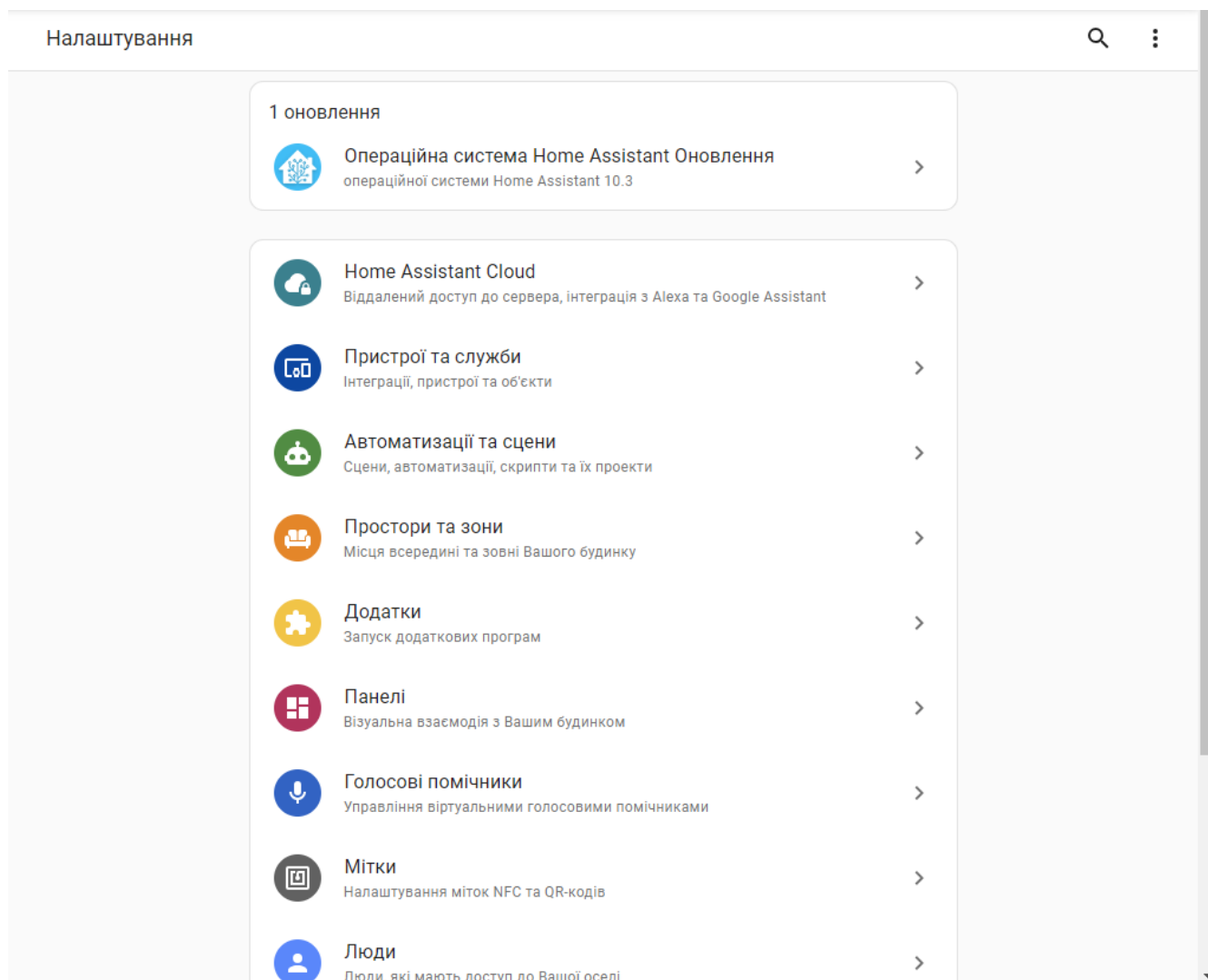


Рисунок 4.18 – Налаштування НА

З’явиться меню інтеграцій. Натискаємо кнопку «Додати інтеграцію» та у вікні брендів вписуємо в пошук «Tuya». З’явиться вікно, в якому необхідно заповнити інформацію щодо інтеграції. (Рисунок 4.19)

The screenshot shows a mobile application window titled "Ваші" (Your) with a help icon (?) and a close icon (X). Below the title is a subtitle: "Налаштування Home Assistant для інтеграції з Tuuya." (Home Assistant settings for integration with Tuuya). The form contains five input fields: 1. "сторона*" (region*) with a dropdown menu showing "Сполучені Штати" (United States). 2. "Ідентифікатор доступу Tuuya IoT*" (Tuuya IoT access identifier*). 3. "Секрет доступу Tuuya IoT*" (Tuuya IoT access secret*) with an eye icon to toggle visibility. 4. "обліковий запис*" (account*). 5. "Пароль*" (password*) with an eye icon to toggle visibility. At the bottom right is a button labeled "підтвердити" (confirm).

Рисунок 4.19 – Вікно для інтеграції Tuuya

Слід зазначити те, що дані для заповнення можна знайти в Tuuya IoT. У вікні потрібно вказати державу, ідентифікатор доступу, секрет доступу, обліковий запис (логін) та пароль. Після цього інтеграція буде підключена та на головній сторінці НА з'являться дані з датчиків. При підключенні інших пристроїв від Tuuya – вони автоматично завантажаться у систему.

4.5 Створення чат-боту

Для зручного керування системою потрібно створити чат-бот Телеграм. Нам знадобиться «BotFather». (Рисунок 4.20).

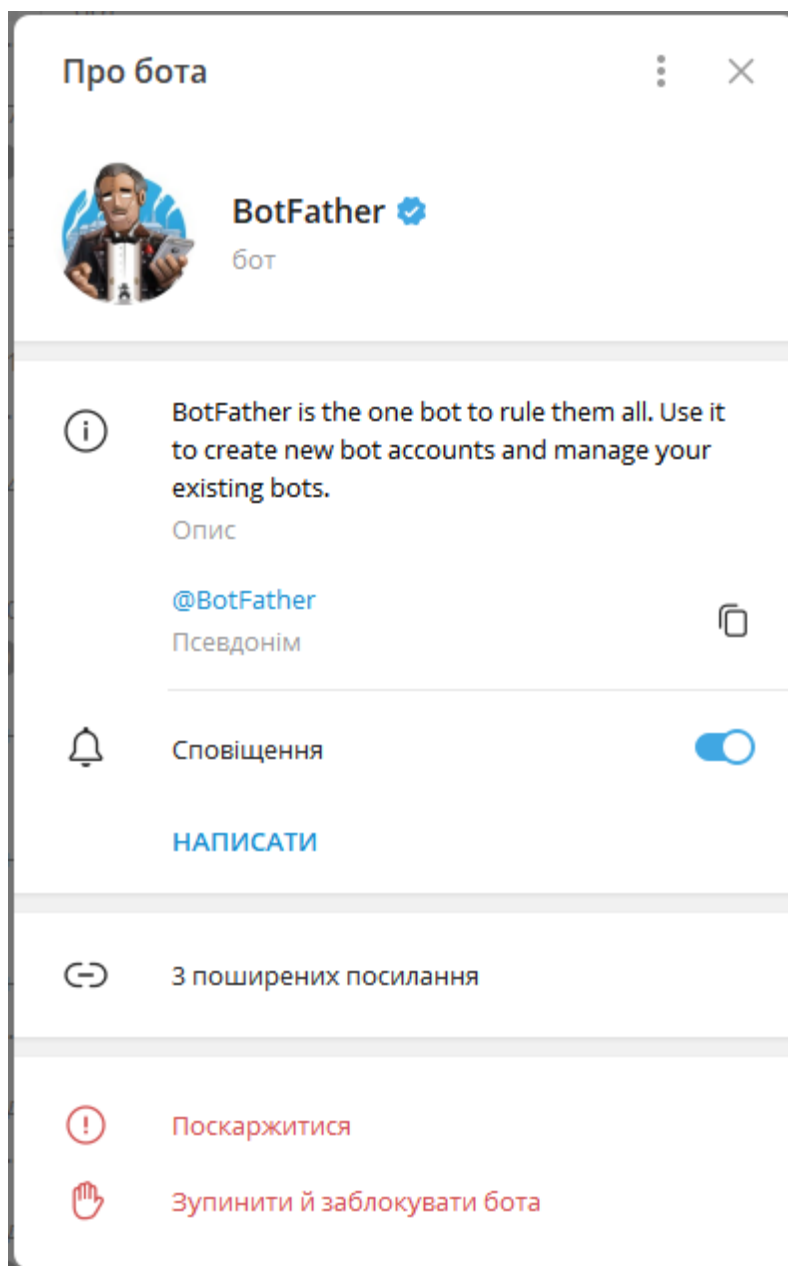


Рисунок 4.20 — BotFather

Етапи створення боту:

– Прописуємо команду «/newbots» для початку створення чат-боту.(Рисунок 4.20)

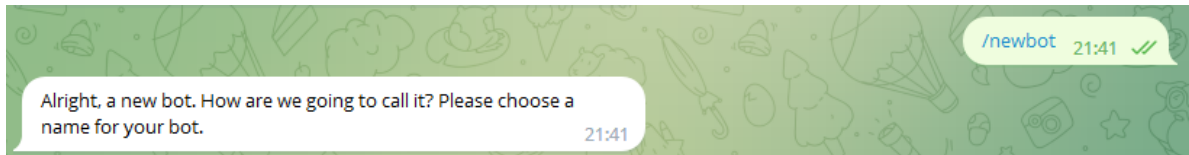


Рисунок 4.20 – Команда «/newbots»

– Нам пропонують вибрати ім'я для бота. Писати його можна як завгодно. Наприклад, «Розумний будиночок». Щоб дати боту вибране ім'я його потрібно просто написати в чат.(Рисунок 4.21)

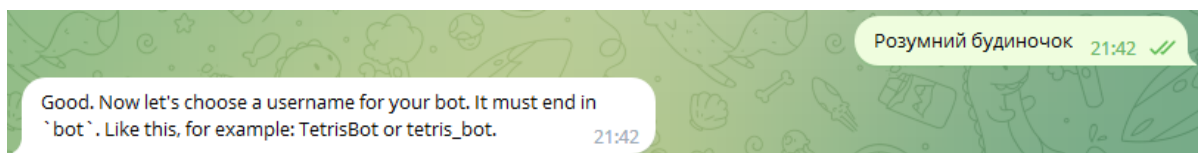


Рисунок 4.21 – Ім'я бота

– Після ім'я потрібно придумати логін. Він відрізняється тим, що його обов'язково потрібно писати латинськими літерами та у нього не повинно бути пробілів. А також обов'язково закінчення «bot» і назва не може повторюватись. Наприклад, «brain_house_bot» (Рисунок 4.22)

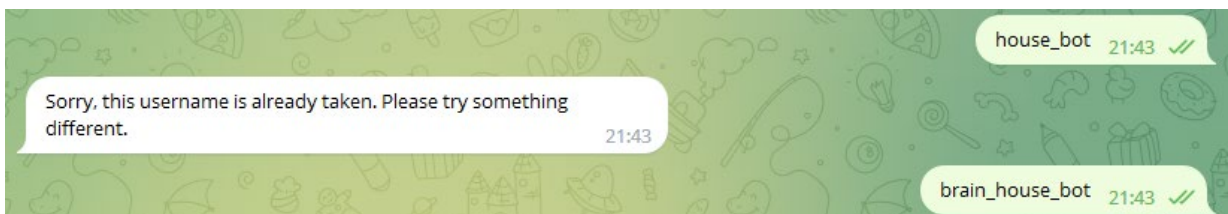


Рисунок 4.22 – Логін бота

Якщо хтось все написав логін, який Ви придумали, то вам повідомлять про це та попросять дати інший.

Виконавши всі вищенаведені пункти, завершуємо процес створення бота, про що також повідомить і ВФ. (Рисунок 4.23)

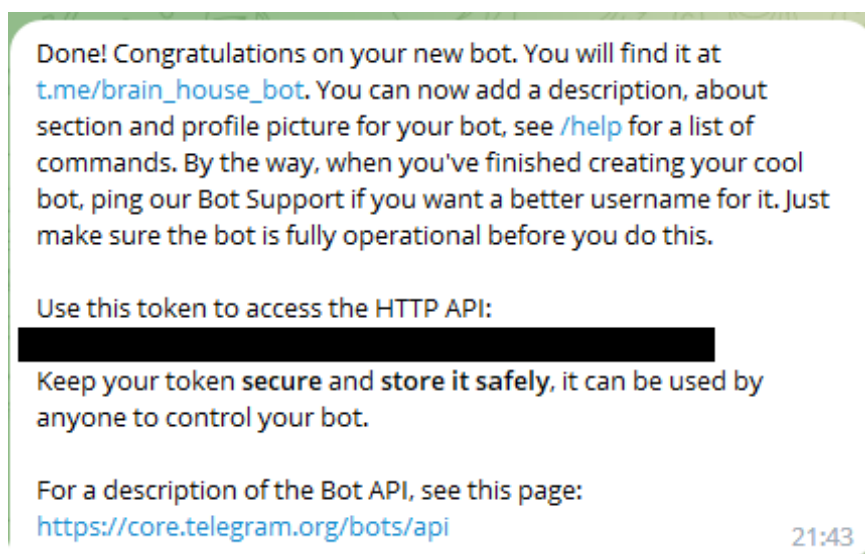


Рисунок 4.23 – Повідомлення про завершення створення бота

У повідомленні також зазначений API ключ, який нам знадобиться для інтегрування в систему НА.

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Налаштування конфігурації НА

Після всіх попередніх етапів потрібно інтегрувати це в чат-бот Телеграму. Система може працювати і без інтеграцій чат-бота, але для зручності користування краще це зробити.

Перед початком інтегрування потрібно налаштувати конфігурацію системи. Для цього потрібно перейти у ядро НА шляхом натискання на кнопку «File editor» або «Редактор файлів». (Рисунок 5.1)

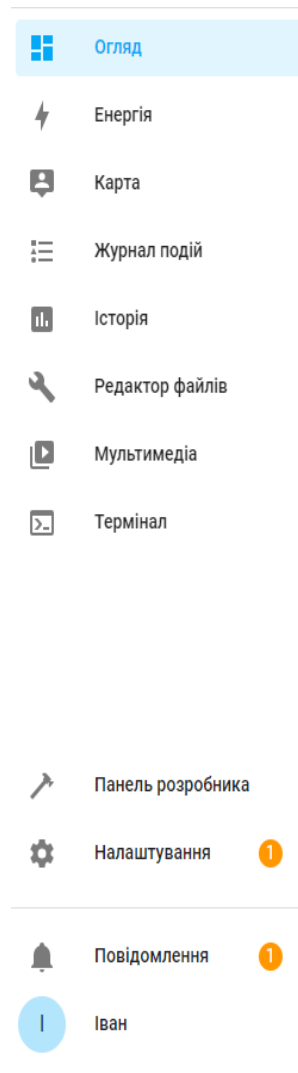


Рисунок 5.1 – Бічна панель

У ядрі НА створюємо папку «includes». Вона використовується для організації зміни та поділу її на логічні блоки. У цій папці можна розмістити файли YAML, які містять фрагменти конфігурації. Основна мета папки «includes» полягає в тому, щоб розбити велику конфігурацію на більш керовані та легко читані частини. В ній створюємо два файли – «telegram.yaml» та «notify.yaml». (Рисунок 5.2)

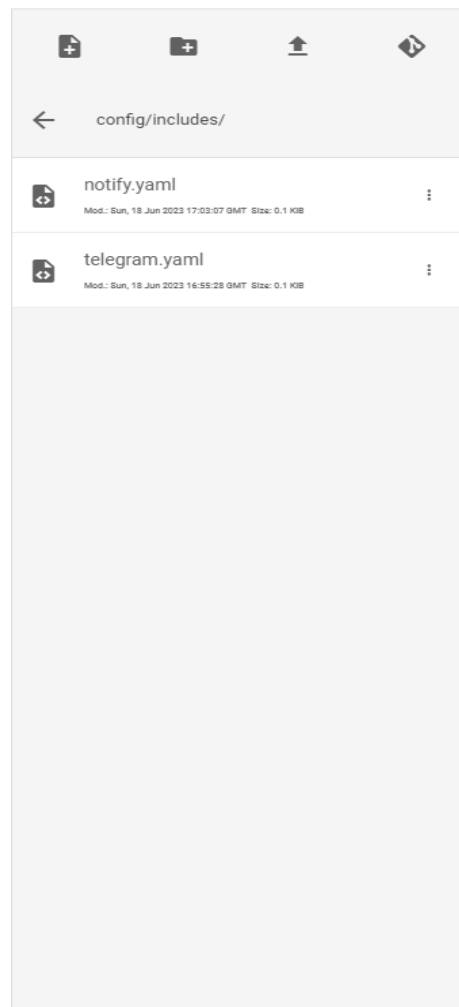


Рисунок 5.2 – Папка «includes»

Перед початком налаштування нам потрібно дізнатися ID чата користувача. Для цього потрібно використати бота під назвою «Show Json Bot». (Рисунок 5.3)

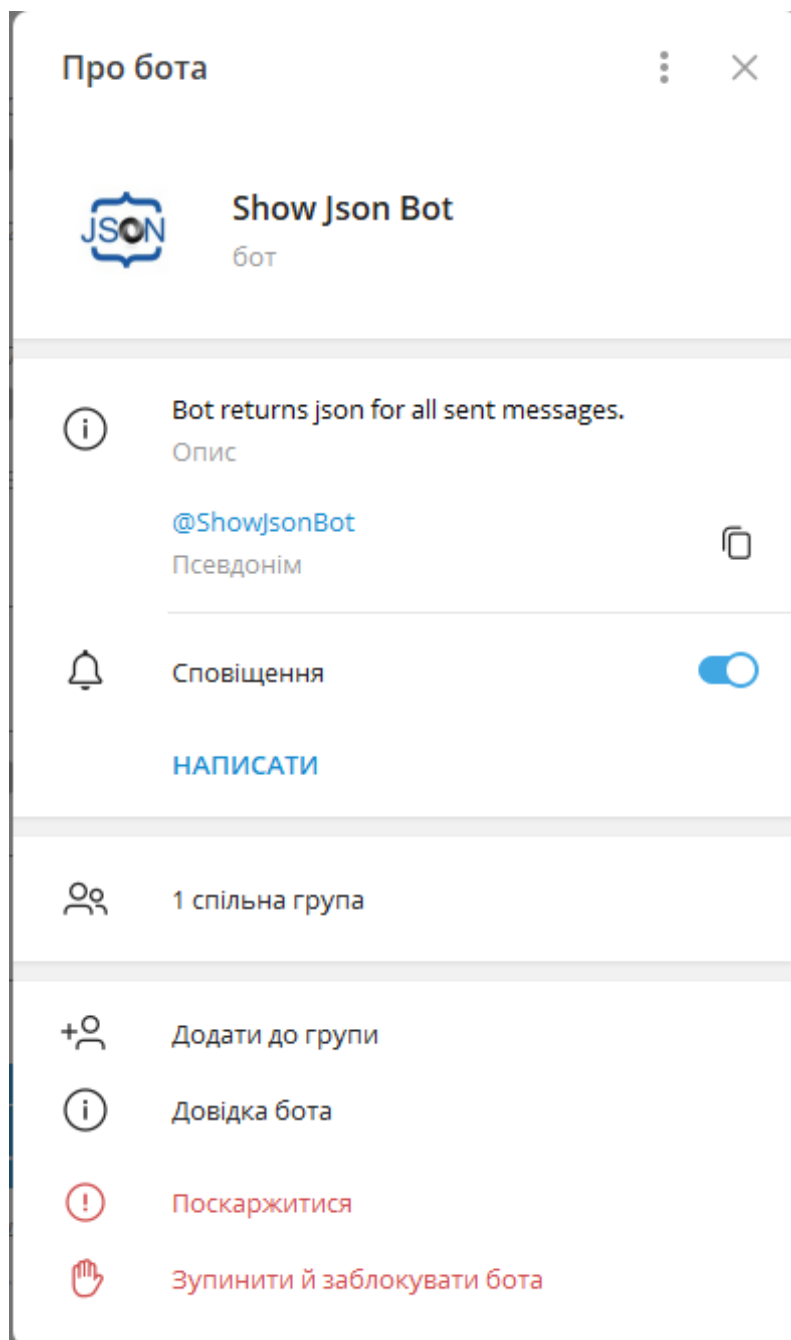
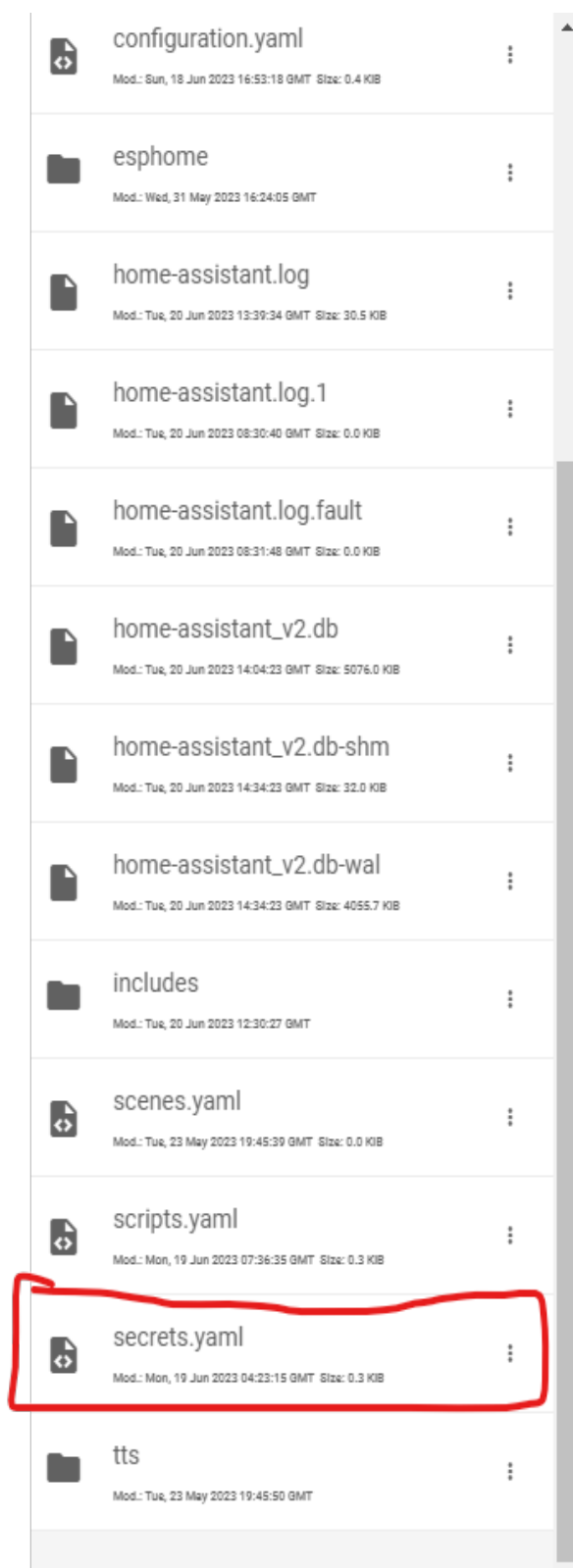


Рисунок 5.3 – бот «Show Json Bot»

Щоб не показувати ID чату користувача та боту – надана можливість вписати ці дані у файл «secrets.yaml». (Рисунок 5.4)



	configuration.yaml Mod.: Sun, 18 Jun 2023 16:53:18 GMT Size: 0.4 KIB	...
	esphome Mod.: Wed, 31 May 2023 16:24:05 GMT	...
	home-assistant.log Mod.: Tue, 20 Jun 2023 13:39:34 GMT Size: 30.5 KIB	...
	home-assistant.log.1 Mod.: Tue, 20 Jun 2023 08:30:40 GMT Size: 0.0 KIB	...
	home-assistant.log.fault Mod.: Tue, 20 Jun 2023 08:31:48 GMT Size: 0.0 KIB	...
	home-assistant_v2.db Mod.: Tue, 20 Jun 2023 14:04:23 GMT Size: 5076.0 KIB	...
	home-assistant_v2.db-shm Mod.: Tue, 20 Jun 2023 14:34:23 GMT Size: 32.0 KIB	...
	home-assistant_v2.db-wal Mod.: Tue, 20 Jun 2023 14:34:23 GMT Size: 4055.7 KIB	...
	includes Mod.: Tue, 20 Jun 2023 12:30:27 GMT	...
	scenes.yaml Mod.: Tue, 23 May 2023 19:45:39 GMT Size: 0.0 KIB	...
	scripts.yaml Mod.: Mon, 19 Jun 2023 07:36:35 GMT Size: 0.3 KIB	...
	secrets.yaml Mod.: Mon, 19 Jun 2023 04:23:15 GMT Size: 0.3 KIB	...
	tts Mod.: Tue, 23 May 2023 19:45:50 GMT	...

Рисунок 5.4 – Файл «secrets.yaml»

При необхідності можна вивести дані за допомогою посилання «!secret». У файлі «telegram.yaml» ми будемо позначати куди відправляти відповідь. (Рисунок 5.5)

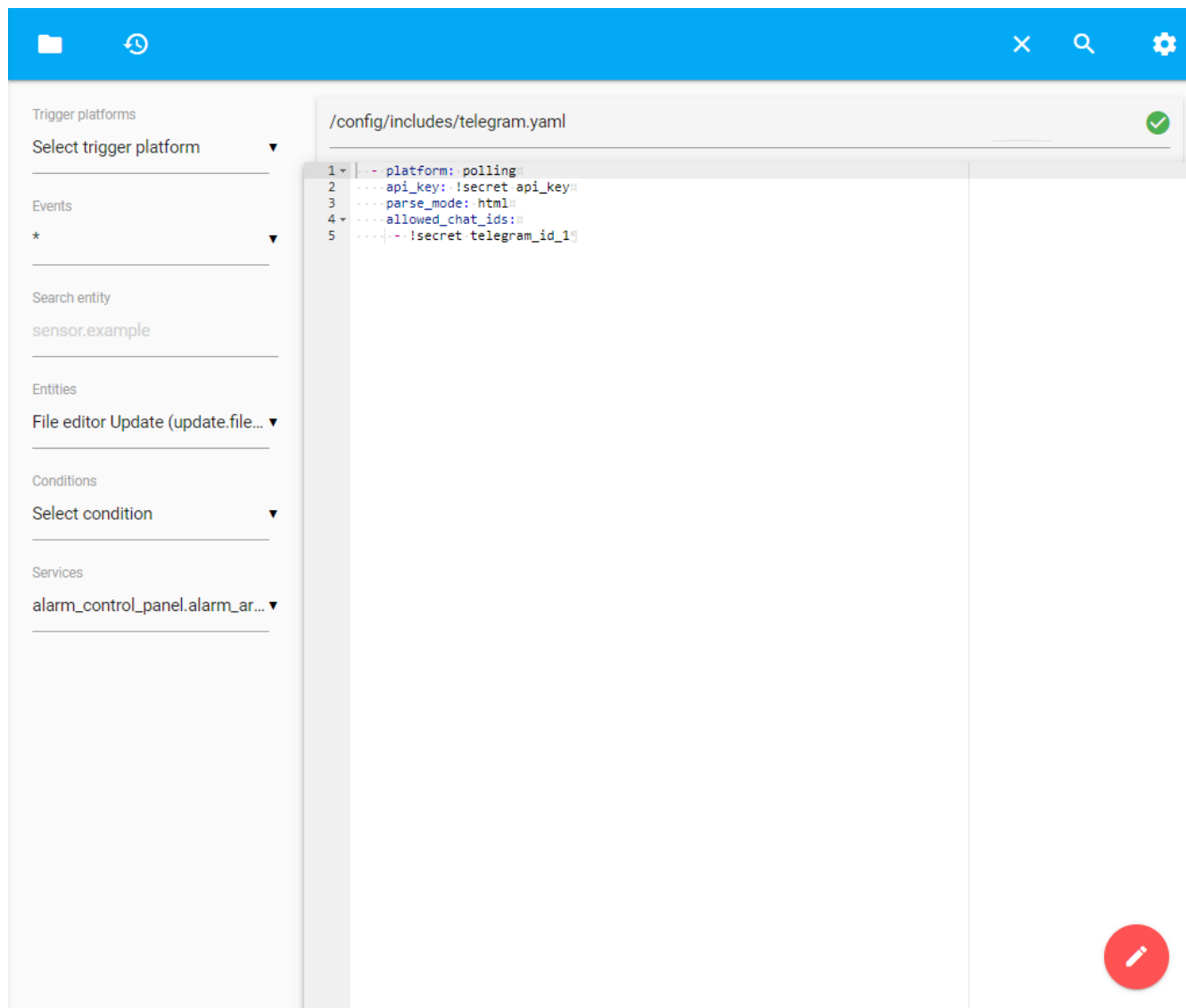


Рисунок 5.5 – Код файлу «telegram.yaml»

У файлі «notify.yaml» ми будемо вказувати користувачів, які будуть користуватися системою. (Рисунок 5.6)

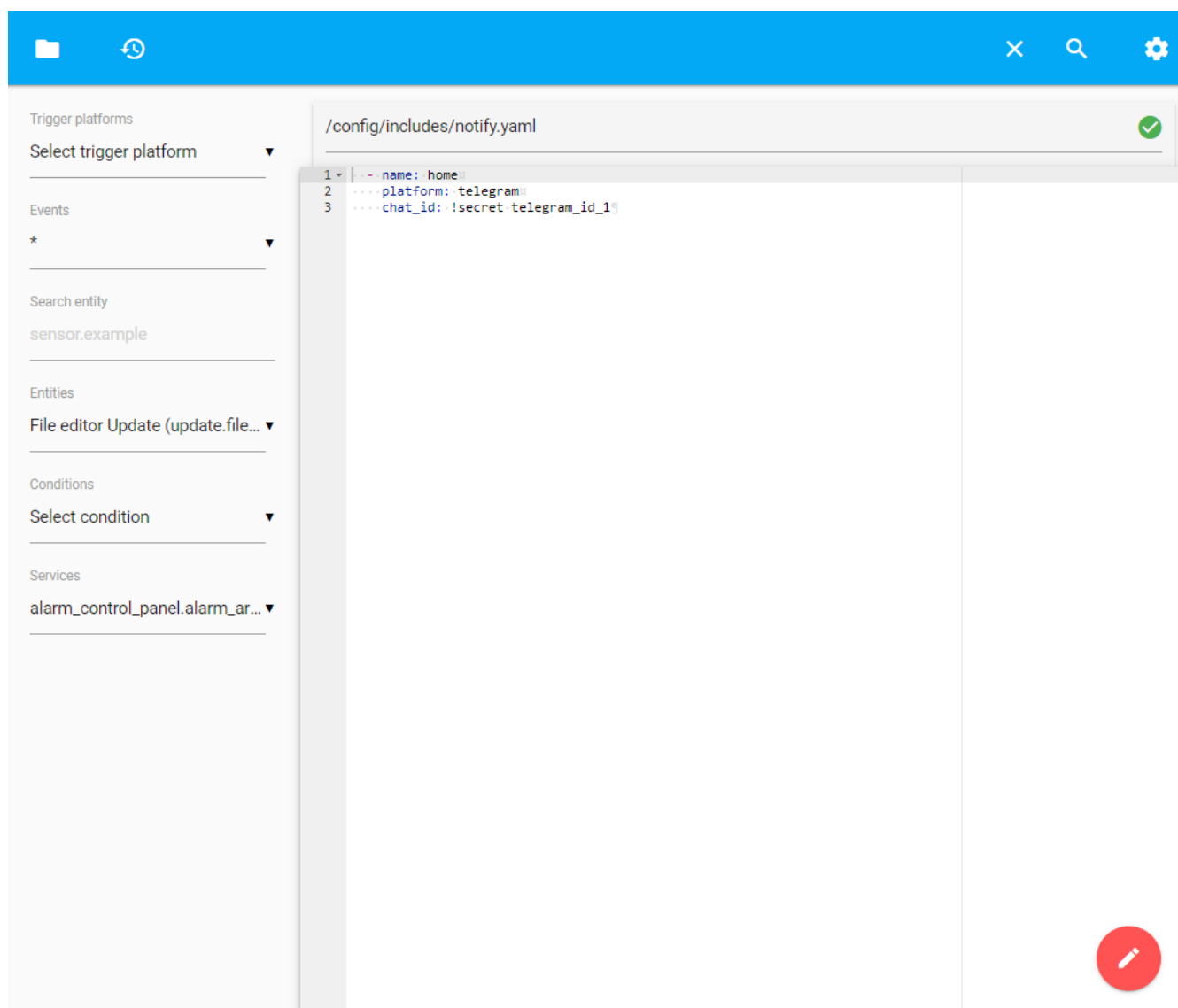


Рисунок 5.6 – Код файлу «notify.yaml»

5.2 Автоматизація служб

Для інтеграції Телеграму і НА залишилось написати рядок коду, який буде визивати служби.

Служби в НА є способом взаємодії з компонентами та пристроями вашого розумного будинку. Вони дозволяють вам керувати різними аспектами вашої системи, відправляти команди на пристрої, запускати автоматизацію та багато іншого. У Home Assistant є безліч вбудованих служб, а також ви можете

створювати власні служби для інтеграції з пристроями та сервісами, які не підтримуються спочатку.

Щоб почати роботу зі службою потрібно перейти у «Панель розробника». (Рисунок 5.7)

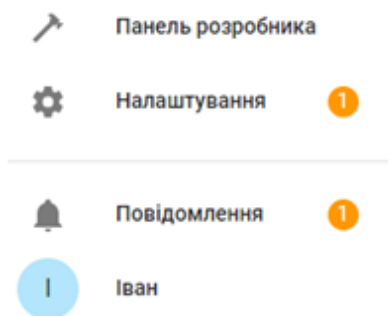


Рисунок 5.7 – Кнопка «Панель розробника»

З'являється меню «Панель розробника» (Рисунок 5.8)

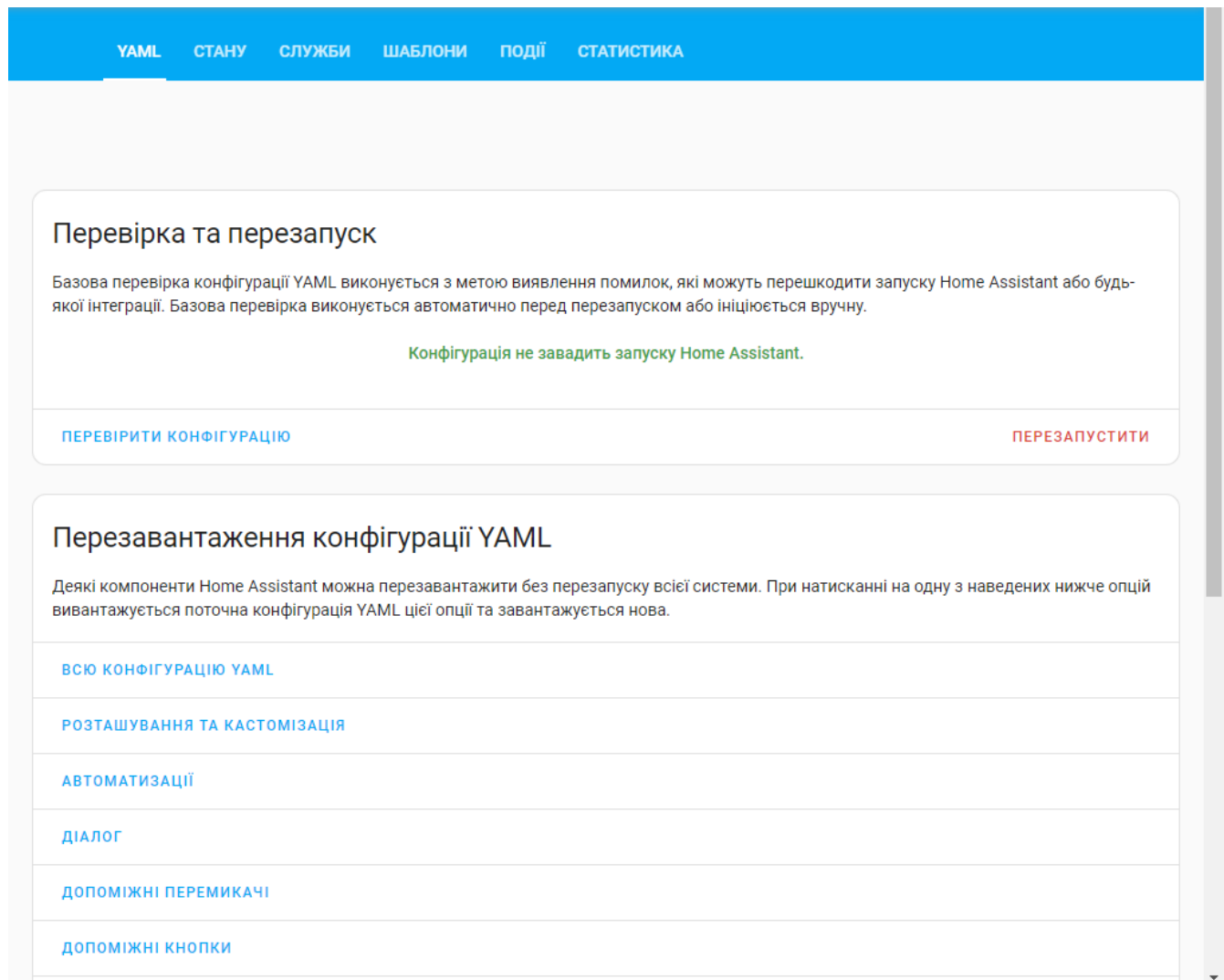


Рисунок 5.8 – Меню «Панель розробника»

Тут надана можливість перевірити всю конфігурацію системи або окремих елементів. У разі виявлення помилки з'явиться повідомлення, в якому буде конкретна інформація щодо помилки, а саме в якому рядку та в чому полягає помилка.

При написанні будь-якої служби в рядку будуть з'являтися підказки на існуючі команди. Для повідомлення використовується служба `notify`. (назва, як була вказана в конфігурації). Наприклад, `notify.home` (Рисунок 5.9)

YAML
СТАНУ
СЛУЖБИ
ШАБЛони
ПОДії
СТАТИСТИКА

Тут Ви можете викликати будь-яку службу Home Assistant зі списку доступних.

Служба
Уведомления: Send a notification with home

×

Надсилає сповіщення за допомогою домашньої служби.

повідомлення

Тіло повідомлення сповіщення.

☐

Назва

Назва сповіщення.

☐

Цільова

Масив цілей для надсилання сповіщень. Додатково в залежності від платформи.

1

☐

Дані

Розширена інформація для повідомлення. Додатково в залежності від платформи.

1

ПЕРЕЙТИ В РЕЖИМ YAML

ВИКЛИКАТИ СЛУЖБУ

Рисунок 5.9 – notify.home

У notify.home є декілька пунктів: повідомлення (message), назва (alias), ціль (target), дані (data). Також є можливість перевести інтерфейс у командний рядок за допомогою кнопки «Перейти в режим Yaml». (Рисунок 5.10)

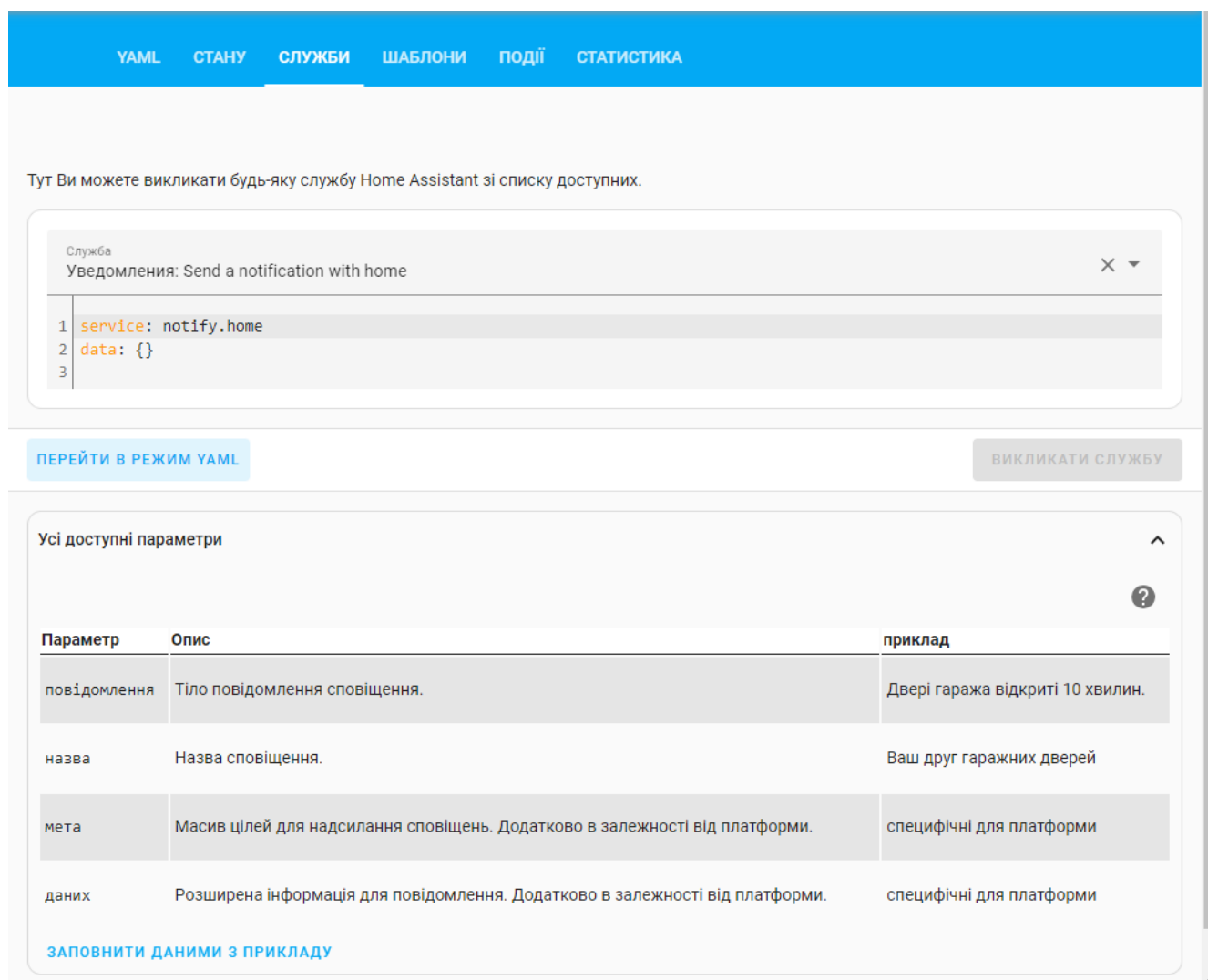


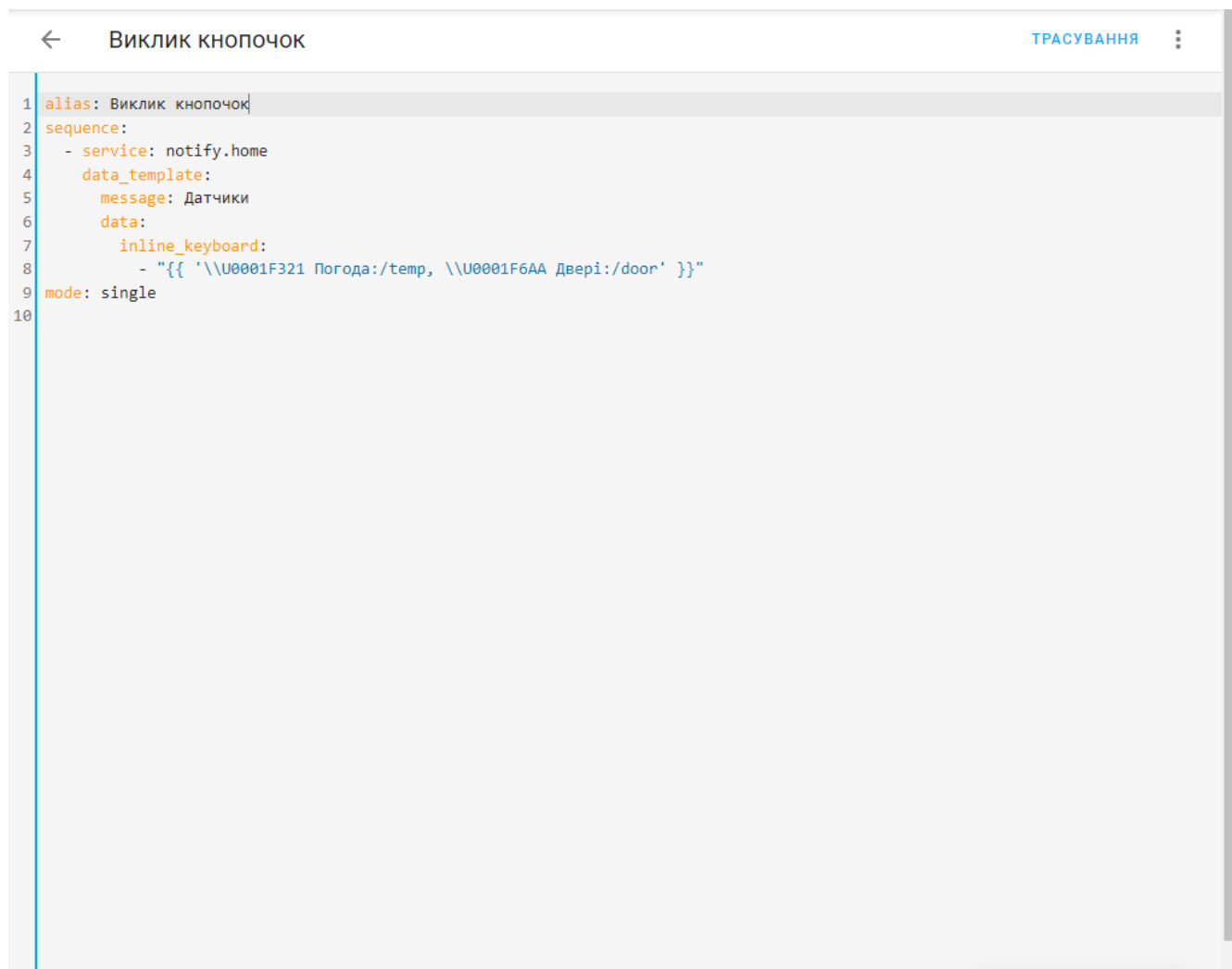
Рисунок 5.10 – Режим Yaml

При написанні служби потрібно буде кожного разу визивати її вручну. Для запобігання даного явища існує автоматизація.

Автоматизація в НА є механізмом для автоматичного виконання завдань та управління вашим розумним будинком на основі певних умов, подій та дій. Вона дозволяє вам створювати правила та сценарії, які автоматично реагують на зміни в системі та виконують певні дії.

Для написання автоматизації потрібно вказати триггер, умову (відстежує стан автоматизації на дану мить) та дію.

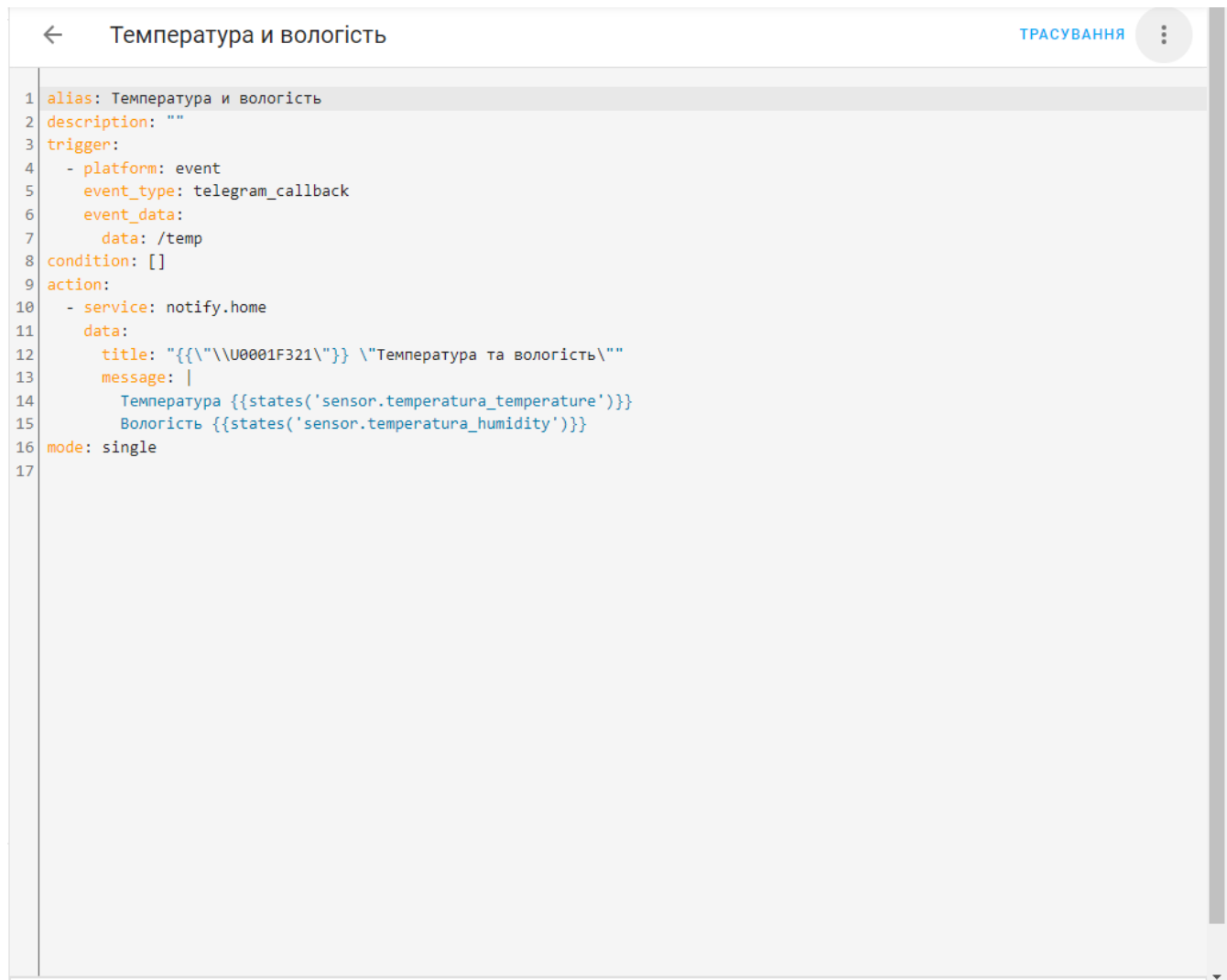
В нашому випадку було зроблено дві автоматизації – для температури та вологості і закритої двері. Для зручності зробимо дві кнопки на кожну із автоматизацій і закрепимо її скриптом, щоб було простіше. (Рисунок 5.11)



```
1 alias: Виклик кнопочок
2 sequence:
3   - service: notify.home
4     data_template:
5       message: Датчики
6       data:
7         inline_keyboard:
8           - "{{ '\U0001F321 Погода:/temp, \U0001F6AA Двері:/door' }}"
9 mode: single
10
```

Рисунок 5.11 – Код скрипту

Тепер потрібно написати автоматизацію для обох датчиків. Для датчику температури та вологості була зроблена кнопка, яка вписує команду «/temp». (Рисунок 5.12)

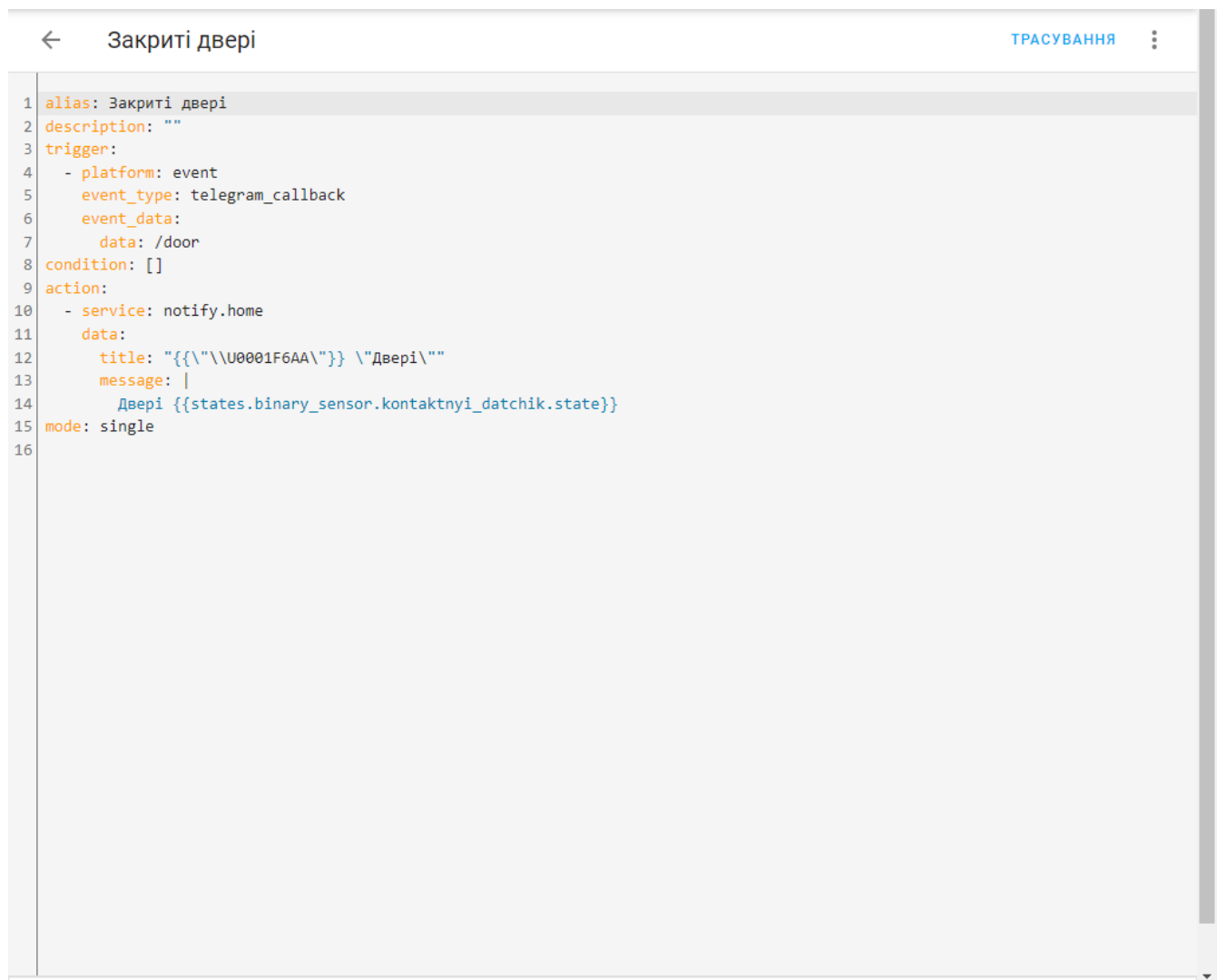


The screenshot shows a code editor window titled "Температура и вологість" (Temperature and humidity). The code is a Home Assistant automation script. It includes a trigger for a telegram callback event, a condition (empty list), and an action to send a notification. The notification title is "Температура та вологість" (Temperature and humidity) and the message contains the current states of the temperature and humidity sensors. The mode is set to "single".

```
1 alias: Температура и вологість
2 description: ""
3 trigger:
4   - platform: event
5     event_type: telegram_callback
6     event_data:
7       data: /temp
8 condition: []
9 action:
10  - service: notify.home
11    data:
12      title: "{{"\U0001F321\}}" "Температура та вологість"
13      message: |
14        Температура {{states('sensor.temperatura_temperature')}}
15        Вологість {{states('sensor.temperatura_humidity')}}
16 mode: single
17
```

5.12 – Код для датчику температури та вологості

Для датчику закритої двері була зроблена кнопка, яка вписує команду «/door». (Рисунок 5.13)



```
1 alias: Закриті двері
2 description: ""
3 trigger:
4   - platform: event
5     event_type: telegram_callback
6     event_data:
7       data: /door
8 condition: []
9 action:
10  - service: notify.home
11    data:
12      title: "{{"\U0001F6AA\}}" \Двері\""
13      message: |
14        Двері {{states.binary_sensor.kontaknyi_datchik.state}}
15 mode: single
16
```

Рисунок 5.13 – Код датчику закритих дверей

Кінцевий вигляд боту зображений на рисунку 5.14



Рисунок 5.14 – Кінцевий вигляд боту

5.3 Безпека системи НА

Безпека є важливим аспектом розумного будинку, і Home Assistant надає кілька функцій та рекомендацій для забезпечення безпеки вашої системи. Ось деякі з них:

- Аутентифікація та авторизація: Home Assistant підтримує різні методи аутентифікації, включаючи логін за допомогою імені користувача та пароля, а також використання зовнішніх систем аутентифікації, таких як OAuth або Home Assistant Cloud. Ви також можете налаштувати різні рівні доступу та авторизації для користувачів, визначати дозволи на керування компонентами та функціями системи.

- HTTPS та шифрування: Рекомендується використовувати HTTPS для захищеного шифрування зв'язку між вашим клієнтським пристроєм та сервером Home Assistant. Home Assistant дозволяє настроїти SSL-сертифікати для використання HTTPS. Це забезпечує конфіденційність та цілісність даних, що передаються між вами та вашою системою.

- Оновлення: Регулярно оновлюйте Home Assistant до останньої версії, щоб отримати всі виправлення помилок та покращення безпеки. Home Assistant має активну спільноту розробників, які постійно працюють над усуненням уразливостей та забезпеченням безпеки системи.

- Зовнішній доступ і віддалене керування: Якщо вам необхідно забезпечити віддалений доступ до системи Home Assistant, рекомендується використовувати VPN (віртуальну приватну мережу) або налаштувати прокидання портів через безпечний маршрутизатор. Це допоможе захистити вашу систему від несанкціонованого доступу ззовні.

- Інтеграція з платформами безпеки: Home Assistant підтримує інтеграцію з різними системами безпеки, такими як системи відеоспостереження, датчики диму та чадного газу, датчики злому та

домофони. Ви можете використовувати ці інтеграції для підвищення безпеки вашого розумного будинку та моніторингу подій безпеки.

- Аудит та логування: Home Assistant надає можливість вести аудит та вести журнал подій у системі. Ви можете налаштувати рівень деталізації логів та моніторити дії, пов'язані з вашою системою Home Assistant.

- Перевірка на вразливості: Регулярно перевіряйте свою систему на наявність уразливостей та дотримуйтесь рекомендацій щодо їх усунення. Це включає оновлення компонентів, перевірку налаштувань безпеки та використання інструментів сканування вразливостей.

ВИСНОВКИ

У ході виконання випускної роботи бакалавра було проведено детальний аналіз сфери діяльності та застосування системи «Розумний будинок». Були розглянуті основні вимоги та функціональні можливості, які повинна мати система.

Проведений відбір програмних засобів, які найкраще відповідають потребам проєкту. Були враховані функціональні вимоги, технічні можливості та вартість програмних засобів.

Проведено обґрунтування вибраних програмних засобів.

Були розглянуті переваги та недоліки кожного засобу, а також відповідність їхніх можливостей потребам проєкту. Також була створена система, яка може мати подальший розвиток. Були враховані всі функціональні вимоги та технічні характеристики.

Після реалізації було проведено налаштування системи та здійснено процес підключення з перевіркою її працездатності та відповідності вимогам проєкту.

На основі проведених розділів випускної роботи можна зробити висновок, що реалізація системи «Розумний будинок» з інтегруванням чат-бота була підготовлена на високому рівні. Були проведені всі необхідні етапи розробки, вибрані відповідні програмні засоби та проведений аналіз економічної ефективності проєкту. Також були враховані аспекти охорони праці для забезпечення безпеки працівників під час реалізації проєкту.

В цілому, випускна робота успішно втілила поставлені цілі та завдання. А також вона має можливість бути вдосконаленим в майбутньому.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Що таке «розумний дім» і навіщо він потрібен? 2022. URL: <https://stylus.ua/uk/articles/528.html>.
2. Розумний дім URL: https://uk.wikipedia.org/wiki/Розумний_дім.
3. Що таке розумний будинок. Платформа безпеки RIPL. 2023. URL: <https://ripl.ua/article/scho-take-rozumnii-budinok>.
4. Управління проектами: навч. посіб. / С. І. Прилипко та ін. Київ : Центр учбової літератури, 2021. 240 с.
5. Kurniawan A. Smart Home Automation with openHAB 3: Getting Started with Java-Based Building Automation. – Berkeley : Apress, 2021. 215 p.
6. Schreiver J. Home Assistant: Customizing Your Home System with Python and YAML. – San Francisco : No Starch Press, 2023. 312 p.
7. Rault S. Building Smart Home Systems with Domoticz and C++. Birmingham : Packt Publishing, 2020. 280 p.
8. Home Assistant: Open source home automation. 2025. URL: <https://www.home-assistant.io>
9. Tuya Smart URL: https://pl.wikipedia.org/wiki/Tuya_smart.
10. Telegram (месенджер) URL: <https://uk.wikipedia.org/wiki/Telegram>
11. Бот (програма) URL: [https://uk.wikipedia.org/wiki/Бот_\(програма\)](https://uk.wikipedia.org/wiki/Бот_(програма)).
12. Tuya IoT Platform: Developer Guide and Cloud Core API Reference Tuya Developer. 2026. URL: <https://developer.tuya.com/en/docs/iot/>
13. Архітектура комп'ютерних систем та мереж Інтернету речей : навч. посіб. /В. В. Пасічник В. А. Резніченко. Львів : Новий Світ-2000, 2023. 288 с.
14. Raspberry Pi URL: https://uk.wikipedia.org/wiki/Raspberry_Pi
15. Home Assistant Documentation. Home Assistant Community, 2025. URL: <https://www.home-assistant.io/docs/>

16. Home Assistant Architecture and Event Loop Home Assistant Developers. 2025. URL: https://developers.home-assistant.io/docs/architecture_index/
17. Інформаційна та кібербезпека в системах Інтернету речей (IoT) : підручник / В. Л. Бурячок та ін. Київ : Гнозіс, 2022. – 340 с.
18. Wi-Fi URL: <https://uk.wikipedia.org/wiki/Wi-Fi>.
19. Tuuya Global IoT Development Platform. Tuuya Inc., 2026. URL: <https://www.tuya.com/>
20. Оліфер К. О., Сумцов К. В. Особливості розробки систем автоматизації житла на базі платформи Home Assistant // *Комп'ютерні науки та інженерія : матеріали Міжнародної науково-технічної конференції*. Київ, 2024. С. 112–115.
21. Kopp C. Automating with Home Assistant: Designing Smart Home Workflows and Custom Integrations using YAML and Jinja2. Offenbach: Rheinwerk Computing, 2024. 312 p.
22. Ткаченко В. М., Коваленко О. С. Дослідження протоколів прикладного рівня в IoT-платформах екосистеми Tuuya Smart. *Вісник ХНУРЕ*. Серія: Комп'ютерні системи та мережі. 2024. Вип. 12. С. 14–21.
23. ДСТУ ISO/IEC 29182-1:2022 Інформаційні технології. Сенсорні мережі: архітектурна платформа. Частина 1. Довідкова архітектура та загальні принципи (ISO/IEC 29182-1:2014, IDT). Київ: ДП «УкрНДНЦ», 2023. 24 с.

ДОДАТОК А Технічне завдання

Вступ

Автоматизація житлового простору є одним із провідних напрямків розвитку сучасних інформаційних технологій та концепції Internet of Things (IoT). Актуальність розробки зумовлена необхідністю створення доступної, гнучкої та енергоефективної системи моніторингу й керування параметрами мікроклімату та безпеки приміщення. Інтеграція платформи домашньої автоматизації з месенджером Telegram через чат-бота дозволяє забезпечити користувача зручним, кросплатформним та захищеним інтерфейсом дистанційного контролю без необхідності встановлення спеціалізованого громіздкого програмного забезпечення на клієнтські пристрої.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломний проєкт бакалавра зі спеціальності 122 «Комп'ютерні науки» на тему «"Розумний будинок" з використанням датчиків температури та інтеграцією чат-бота», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 07 квітня 2026 року.

A.2 Призначення розробки

Програмно-апаратна система призначена для:

- автоматизації збору даних про стан навколишнього середовища (температури) та безпеки (стану дверей) всередині приміщення;
- централізованого інтелектуального керування підключеними побутовими приладами та смарт-пристроями;

□ оперативного віддаленого інформування користувача про критичні зміни параметрів або позаштатні ситуації через чат-бота месенджера Telegram.

A.3 Вимоги до програмного продукту

A.3.1 Вимоги до функціональних характеристик

Система повинна стабільно забезпечувати виконання таких основних функцій:

- 1. Збір та обробка даних:** Безперервне отримання телеметрії від бездротових датчиків температури та датчиків відчинення дверей.
- 2. Централізоване керування:** Інтеграція і координація роботи підключених виконавчих пристроїв та реле через локальний сервер автоматизації.
- 3. Взаємодія через чат-бот:** Дистанційне керування компонентами «Розумного будинку» (увімкнення/вимкнення, зміна цільової температури) за допомогою текстових команд та інтерактивного меню у Telegram.
- 4. Сповіщення користувача:** Автоматичне генерування та надсилання тривожних повідомлень у чат-бот у разі виходу показників за межі встановлених лімітів або при фіксації несанкціонованого доступу (відкриття дверей).
- 5. Модульність та розширюваність:** Можливість додавання нових периферійних пристроїв, виконавчих механізмів та додаткових датчиків без модифікації базового архітектурного ядра системи.

A.3.2 Вимоги до надійності

□ Безперервне функціонування серверної частини системи в режимі 24/7 (цілодобово).

□ Автоматичне відновлення зв'язку з кінцевими IoT-пристроями та зовнішніми хмарними сервісами після збоїв у мережі електроживлення або тимчасового зникнення Інтернет-з'єднання.

□ Програмна фільтрація та валідація вхідних даних від датчиків для виключення хибних спрацьовувань алгоритмів сповіщення.

А.3.3 Вимоги до умов експлуатації

□ Серверна частина системи (Home Assistant) має розгортатися на базі мікрокомп'ютера і функціонувати в автономному режимі всередині приміщення при кімнатній температурі.

□ Для повноцінного функціонування системи обов'язкова наявність локальної бездротової мережі Wi-Fi з постійним стабільним доступом до Інтернету для взаємодії з екосистемою Tuuya Smart та Telegram Bot API.

А.3.4 Вимоги до складу та параметрів технічних засобів

Для розгортання та безперебійної експлуатації системи необхідний такий мінімальний комплект апаратного забезпечення:

□ Одноплатний комп'ютер Raspberry Pi 4 (або сумісний обчислювальний модуль для сервера керування).

□ Маршрутизатор бездротової мережі (Wi-Fi роутер) із підтримкою протоколів зв'язку IEEE 802.11 b/g/n.

□ Розумні пристрої та датчики: бездротовий датчик температури, датчик відкриття дверей, що підтримують інтеграцію з платформою Tuuya Smart / Home Assistant.

□ Мобільний пристрій або персональний комп'ютер користувача із встановленим клієнтським додатком Telegram.

A.3.5 Вимоги до програмної сумісності

Система має бути побудована із використанням таких програмних засобів, платформ та мов програмування:

□ **Операційна система сервера:** Linux (Home Assistant OS / Debian-based OS).

□ **Ядро автоматизації:** Платформа з відкритим вихідним кодом Home Assistant.

□ **Мова розробки чат-бота:** Python (із використанням спеціалізованих бібліотек для взаємодії з Telegram API).

□ **Хмарні платформи та інтерфейси (API):** Екосистема TuYa IoT Platform, Telegram Bot API (створення бота через BotFather).

A.3.6 Вимоги до маркування та пакування

Розробка є програмно-орієнтованою. Вихідні тексти програмного коду модулів чат-бота мають містити детальні коментарі українською мовою із зазначенням призначення функцій, класів та змінних. Кінцева робоча конфігурація сервера Home Assistant, скрипти автоматизації та файли коду чат-бота мають зберігатися у вигляді структурованого архіву (резервної копії) для швидкого розгортання на новому обладнанні.

A.4 Вхідні та вихідні дані

Вхідними даними для системи є інформація, що надходить із зовнішніх сенсорів, платформ автоматизації, а також безпосередньо від користувача через інтерфейс взаємодії. До них належать:

Телеметрія від бездротових IoT-пристроїв (TuYa Smart):

□ Поточні значення температури в приміщенні (числовий формат, наприклад, 22.5 °C).

□ Дискретні сигнали від датчиків відчинення дверей (логічний формат: on/off або true/false).

□ Поточний стан елементів живлення сенсорів (у відсотках від 0% до 100%).

□ Мережевий статус підключеного обладнання (online/offline).

2. Запити від користувача (Telegram-бот):

□ Текстові команди керування, введені вручну (наприклад, /start, /status, /help).

□ Ідентифікатори зворотного зв'язку (Callback Data) при натисканні інтерактивних інлайн-кнопок меню.

□ Унікальний ідентифікатор користувача (user_id / chat_id) для авторизації в системі.

3. Конфігураційні параметри:

□ Встановлені користувачем температурні пороги для спрацьовування сценаріїв автоматизації.

□ Ключі доступу до хмари TuYa та унікальний токен Telegram Bot API.

Вихідними даними системи є керуючі сигнали, що надсилаються на виконавчі пристрої, а також інформаційні масиви, що транлюються користувачеві. До них належать:

1. Керуючі сигнали для виконавчих пристроїв:

□ Цифрові команди увімкнення/вимкнення живлення розумних реле, розеток або кліматичних приладів.

2. Інформаційні повідомлення для користувача:

- Текстові звіти про поточний стан мікроклімату та безпеки.
- Оновлені блоки інтерактивних інлайн-кнопок для навігації по меню бота.

- Інструкції та підказки щодо використання інтерфейсу.

3. Тривожні та сервісні сповіщення:

- Миттєві повідомлення у чат при виході температури за межі норми або при фіксації відкриття дверей у режимі охорони.

4. Системні логи:

- Записи у журналі подій сервера Home Assistant про стан датчиків та хронологію виконання команд.

ДОДАТОК Б Фрагмент коду програми

Б.1 Код скрипту Telegram-бота та ініціалізації інтерфейсу

alias: Виклик кнопочок

sequence:

- service: notify.home

data_template:

message: Датчики

data:

inline_keyboard:

- "{ { '\U0001F321 Погода:/temp, \U0001F6AA Двері:/door' } }"

mode: single

Б.2 Код датчика температури та вологості

alias: Температура и вологість

description: ""

trigger:

- platform: event

event_type: telegram_callback

event_data:

data: /temp

condition: []

action:

- service: notify.home

data:

title: "{ { '\U0001F321\'' } } \"Температура та вологість\''"

```

message: |
    Температура {{states('sensor.temperatura_temperature')}}
    Вологість {{states('sensor.temperatura_humidity')}}

```

```
mode: single
```

Б.3 Код датчику закритої двері

```

alias: Закриті двері
description: ""
trigger:
  - platform: event
    event_type: telegram_callback
    event_data:
      data: /door

condition: []

action:
  - service: notify.home
    data:
      title: "{{"\U0001F6AA\}}" \\"Двері\\"""
      message: |
        Двері {{states.binary_sensor.kontaktnyi_datchik.state}}

mode: single

```

Б.4 Код глобальних конфігураційних файлів ядра Home Assistant

Б.4.1 Базова конфігурація системи (configuration.yaml)

ГОЛОВНИЙ КОНФІГУРАЦІЙНИЙ ФАЙЛ СЕРВЕРА
(configuration.yaml)

Забезпечує інтеграцію модулів та ініціалізацію внутрішньої машини станів

homeassistant:

name: "Розумний Будинок"

latitude: !secret home_latitude

longitude: !secret home_longitude

elevation: 105

unit_system: metric

time_zone: Europe/Kyiv

external_url: !secret ha_external_url

internal_url: !secret ha_internal_url

Активація стандартного набору компонентів ядра

default_config:

Конфігурація бази даних для збереження історії вимірювань датчиків

recorder:

purge_keep_days: 7

db_url: sqlite:///config/home-assistant_v2.db

include:

domains:

- sensor

- binary_sensor

entities:

- sensor.temperatura_temperature
- sensor.temperatura_humidity
- binary_sensor.kontaktnyi_datchik

Налаштування HTTP-шлюзу для безпечного приймання зовнішніх запитів

http:

use_x_forwarded_for: true

trusted_proxies:

- 172.30.33.0/24

Інтеграція модульних файлів з папки includes (Розділ 5.1)

telegram_bot: !include includes/telegram.yaml

notify: !include includes/notify.yaml

automation: !include automations.yaml

script: !include scripts.yaml

Б.4.2 Код конфігурації шлюзу Webhook

МОДУЛЬ ІНТЕГРАЦІЇ TELEGRAM BOT API (includes/telegram.yaml)

Налаштування асинхронного веб-хуку для взаємодії з BotFather

- platform: webhooks

api_key: !secret telegram_bot_token

allowed_chat_ids:

- !secret telegram_allowed_chat_id_user1

Налаштування таймаутів очікування серверної черги

parse_mode: html

proxy_url: ""

Б.4.3 Код реєстрації сервісів сповіщення

МОДУЛЬ РЕЄСТРАЦІЇ СЛУЖБ СПОВІЩЕННЯ
(includes/notify.yaml)

Створення об'єкта notify.home для маршрутизації відповідей

- platform: telegram

name: home

chat_id: !secret telegram_allowed_chat_id_user1

Б.4.4 Файл безпеки та прихованих токенів

Криптографічний захист персональних токенів та адрес

home_latitude: 47.8388

home_longitude: 35.1396

ha_external_url: "https://secure-smart-home-bot.duckdns.org"

ha_internal_url: "http://192.168.1.105:8123"

telegram_bot_token:

"7483920155:AAFkldjf93820Fkdsj928301_kdsjFHks8"

telegram_allowed_chat_id_user1: 987654321

Б.5 Код розширених системних сенсорів обробки даних

Б.5.1 Шаблони віртуальних сенсорів стану приміщення

```
# Створення додаткових аналітичних об'єктів для бота
- platform: template
  sensors:
    climat_comfort_status:
      friendly_name: "Рівень кліматичного комфорту"
      value_template: >-
        {% set temp = states('sensor.temperatura_temperature') | float %}
        {% set hum = states('sensor.temperatura_humidity') | float %}
        {% if temp >= 18.0 and temp <= 24.0 and hum >= 40.0 and hum <=
60.0 %}
          Оптимальний
        {% else %}
          Потребує регулювання
        {% endif %}

- platform: statistics
  name: "Максимальна температура за добу"
  entity_id: sensor.temperatura_temperature
  state_characteristic: value_max
  max_age:
    hours: 24

- platform: statistics
  name: "Мінімальна температура за добу"
  entity_id: sensor.temperatura_temperature
```

```
state_characteristic: value_min
max_age:
  hours: 24
```

Б.5.2 Шаблони бінарних сенсорів безпеки периметра

```
# Налаштування логічних перевірок для вхідної групи дверей
- platform: template
sensors:
  home_perimeter_alarm:
    friendly_name: "Порушення контуру охорони"
    device_class: safety
    value_template: >-
      {{ is_state('binary_sensor.kontaktnyi_datchik', 'on') }}
    attribute_templates:
      last_event_time: "{{ now().strftime('%Y-%m-%d %H:%M:%S') }}"
```

Б.6 Код автоматизацій системи безпеки та оповіщення

```
# Додаткові сценарії автоматичного фонового моніторингу
- id: '1720000000003'
  alias: Автоматичне попередження про відкриття
  description: "Надсилення тривоги у разі зміни стану вікон чи дверей"
  trigger:
    - platform: state
      entity_id: binary_sensor.kontaktnyi_datchik
      from: 'off'
      to: 'on'
```

condition: []

action:

- service: notify.home

data:

title: "☐ ТРИВОГА: Двері відчинено!"

message: |

Увага! Зафіксовано відкриття вхідної групи дверей приміщення.

Дата події: {{ now().strftime('%d.%m.%Y') }}

Час події: {{ now().strftime('%H:%M:%S') }}

mode: single

- id: '1720000000004'

alias: Сповіщення про низький заряд батареї

description: "Контроль автономних джерел живлення периферійних пристроїв"

trigger:

- platform: numeric_state

entity_id: sensor.temperatura_battery

below: 15

condition: []

action:

- service: notify.home

data:

title: "☐ Системне попередження"

message: "Увага! Рівень заряду батареї датчика температури впав нижче 15%. Необхідно провести заміну елементів живлення."

mode: single

ДОДАТОК В Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Створення інтелектуальної системи
""Розумний будинок"" з використанням
датчиків руху та температури

Студент, групи КНТ-213сп

Іван МІЩЕНКО

Керівник проєкту, к.т.н, доцент

Олександр СТЕПАНЕНКО

Об'єкт, предмет, мета

Об'єкт дослідження – методи, алгоритми і комп'ютерні обчислення та їх застосування для створення програмних систем та віддаленого моніторингу параметрів середовища на базі концепції Інтернету речей (IoT).

Предмет дослідження – архітектурні рішення та програмно-апаратні засоби інтеграції локального сервера домашньої автоматизації Home Assistant з месенджером Telegram для забезпечення дистанційного контролю.

Мета роботи – підвищення комфорту, безпеки та ефективності керування параметрами мікроклімату приміщення шляхом розробки і налаштування системи «Розумний будинок» з інтегрованим кросплатформним чат-ботом Telegram.

Рисунок В.2 — Слайд 2

Обґрунтування вибору технологій

Критерій порівняння	Home Assistant (HA)	openHAB	Domoticz	Tuya Smart / Smart Life
Мова програмування ядра	Python	Java	C++	Java
Основна архітектура	Локальна (Local-first)	Локальна (Local-first)	Локальна (Local-first)	Хмарна (Cloud-based)
Формат конфігурації	YAML / Веб-інтерфейс	Специфічні файли конфігурації (.items, .rules)	Скрипти Lua / Blockly / Веб-інтерфейс	Мобільний додаток (No-code)
Споживання ресурсів	Помірне (рекомендовано від 1-2 ГБ RAM)	Високе (через віртуальну машину Java JVM)	Мінімальне (працює навіть на 256 МБ RAM)	Мінімальне на боці клієнта (все обчислюється в хмарі)
Кількість інтеграцій	Понад 2500+	Близько 400+	Обмежена (потребує ручного налаштування)	Обмежена лише пристроями партнерів Tuya
Взаємодія з роботом (API)	Повна підтримка Webhooks, REST API, WebSockets	REST API	HTTP API / MQTT	Обмежений Tuya IoT Cloud API

Рисунок В.3 — Слайд 3

Архітектура системи

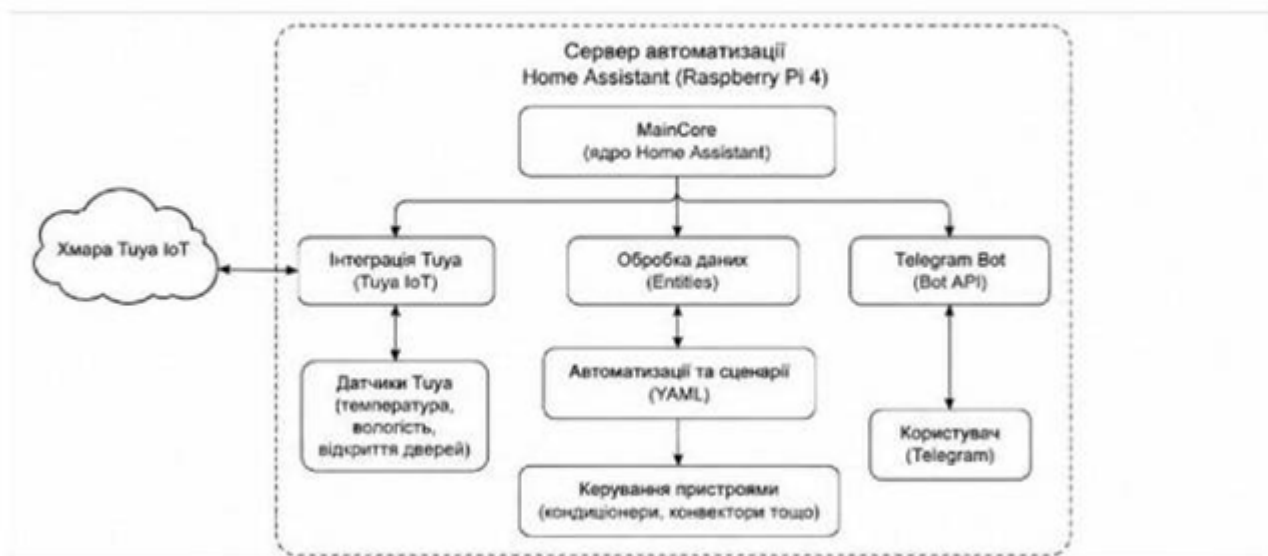


Рисунок В.4 — Слайд 4

Логічна структура програмного забезпечення сервера



Діаграма логічної структури програмного забезпечення сервера

Меню інтеграцій

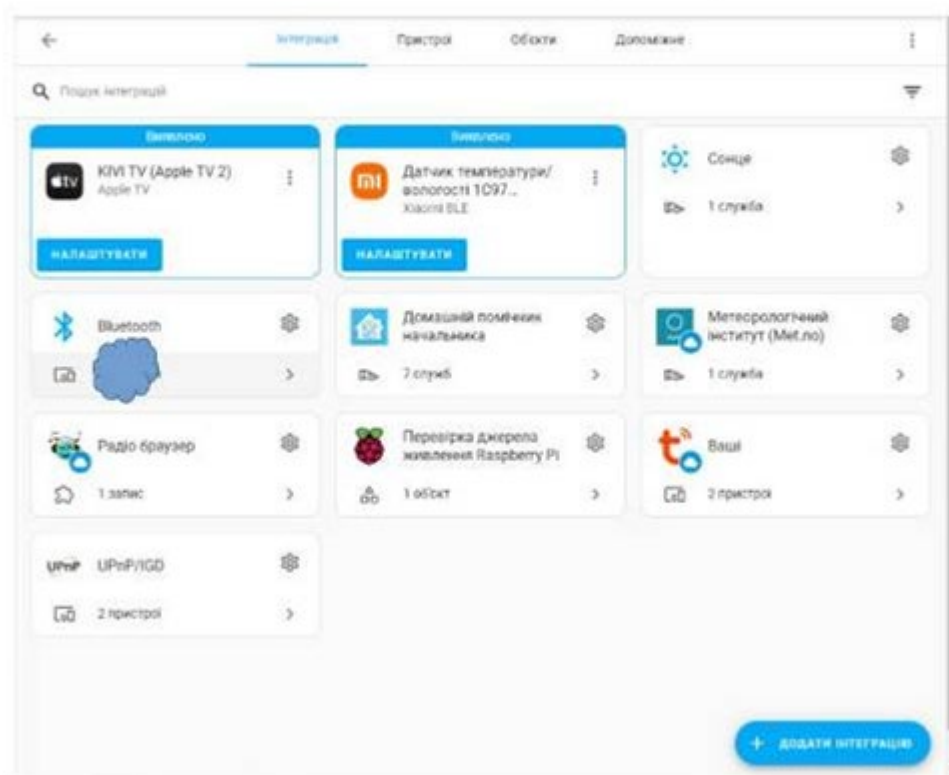


Рисунок В.6 — Слайд 6

“BotFather”

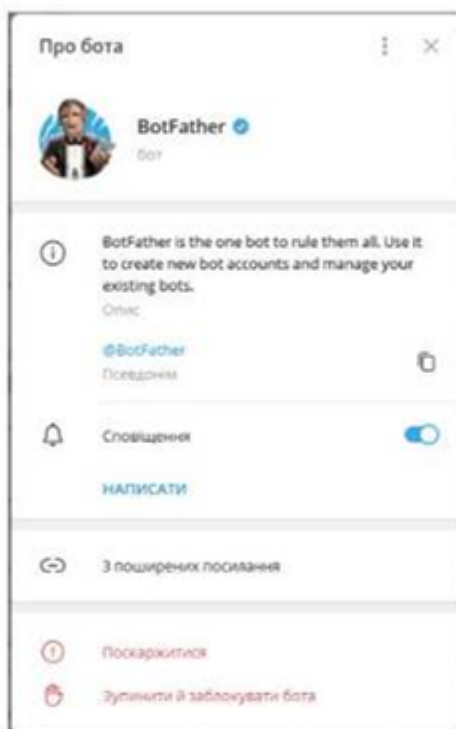


Рисунок В.7 — Слайд 7

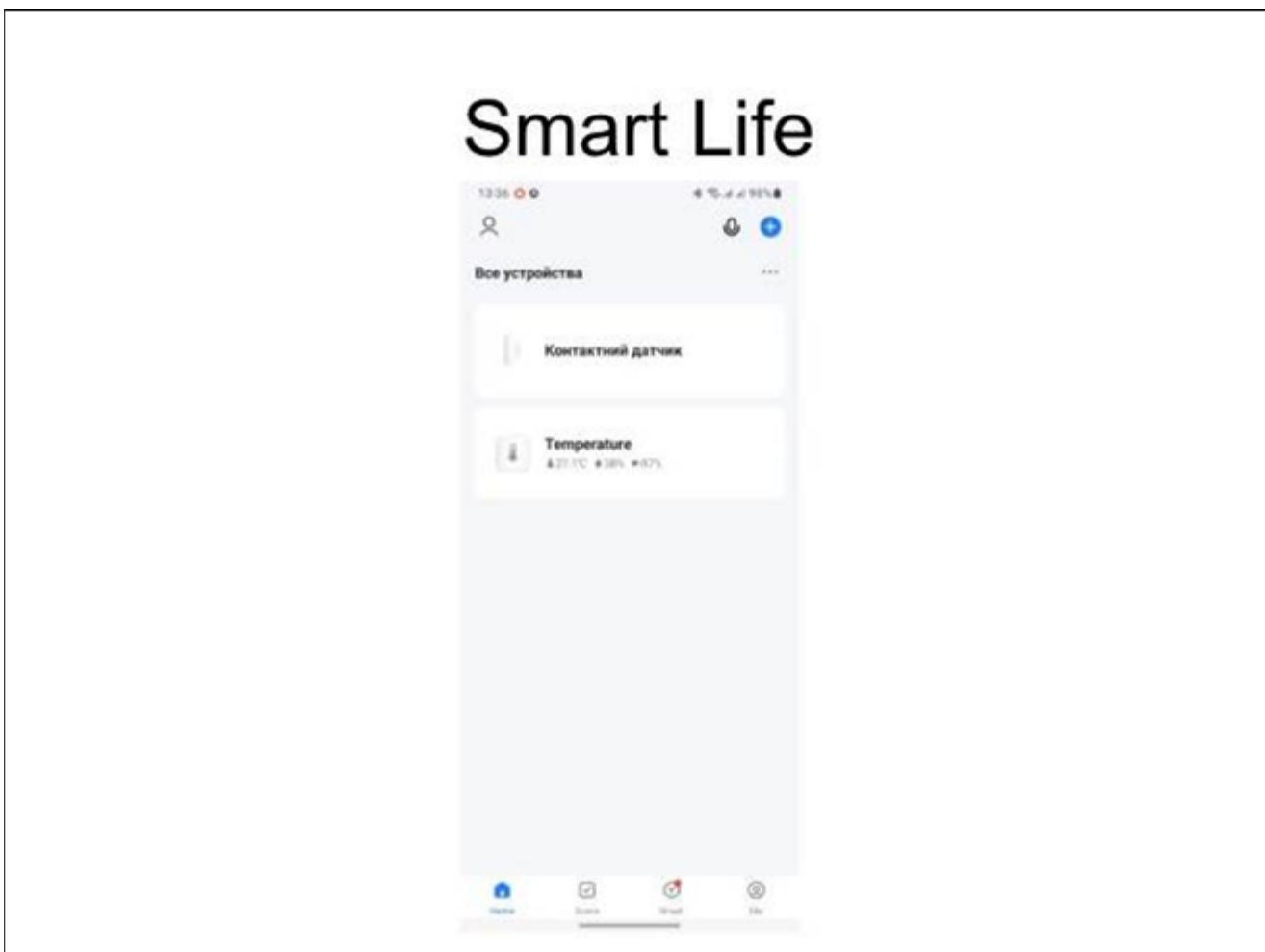


Рисунок В.8 — Слайд 8

Чат з ботом



Рисунок В.9 — Слайд 9

Висновки

У ході виконання випускної роботи бакалавра було проведено детальний аналіз сфери діяльності та застосування системи «Розумний будинок». Були розглянуті основні вимоги та функціональні можливості, які повинна мати система.

Проведений відбір програмних засобів, які найкраще відповідають потребам проєкту. Були враховані функціональні вимоги, технічні можливості та вартість програмних засобів.

Проведено обґрунтування вибраних програмних засобів.

Були розглянуті переваги та недоліки кожного засобу, а також відповідність їхніх можливостей потребам проєкту. Також була створена система, яка може мати подальший розвиток. Були враховані всі функціональні вимоги та технічні характеристики.